September 2004

SERVO MAGAZINE

Robotics — Past • Present • Future

CORDIC ALGORITHMS

FLEXING YOUR NANOMUSCLES

CASTING AND MOLD MAKING

TETSUJIN 2004:

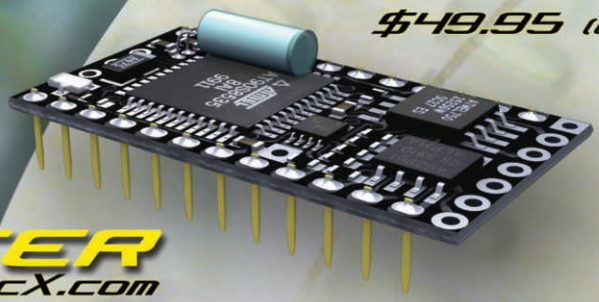
From Blueprint TO REALITY...

ANYTHING THEY CAN DO...

WE DO...

BASICX24™

$49.95 (Qty 1)



...FASTER

WWW.BASICX.COM

Executing 65,000 lines of Basic code per second the BasicX-24 is the KING of Basic programmable microcontrollers.

400 bytes RAM.
32K User program area.
19 I/O lines with 8 10Bit ADC's.
Real multitasking and Serial UARTs.

...SMALLER

WWW.SITEPLAYER.COM

Siteplayer is a true stand-alone mini web server.

Super easy to use.
Standard RJ-45 network interface.
Control or monitor anything over the web.

SITEPLAYER™

$29.95 (Qty 1)

2x16 SERIALLCD™

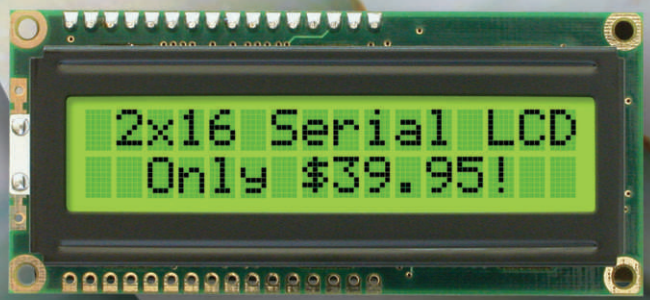
$39.95 (Qty 1)

...BETTER

WWW.BASICX.COM

High quality serial 2x16 LCD with backlight

Easy to use.
2400 & 9600 Baud support
Software controllable backlight and contrast.



Circle #60 on the Reader Service Card.

NetMedia®

NETMEDIA INC. 10940 NORTH STALLARD PLACE TUCSON ARIZONA 85737

WWW.NETMEDIA.COM 520. 544. 4567

CIRCUIT BOARD solutions for your robotic needs

Need FREE design software?

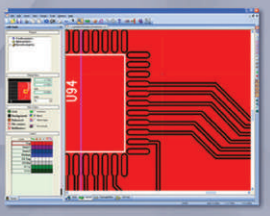
FREE Download the new version at www.pcb123.com

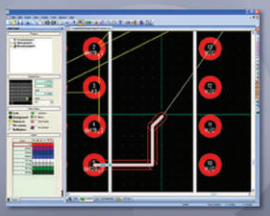
PCB123®

VERSION 2.0

PCB123 Provides the Complete Circuit Board Solution, Design Through Order

PCB123 is a **FREE** layout and schematic design software tool that quotes your board while you design!





New / Enhanced Features:

- → Import Net lists from multiple CAD vendors
- → 145,000+ parts library
- → 2, 4, and 6 layer support
- → Fast, accurate DRC
- → Copper pour with net connectivity
- → Fine pitch autorouter with fanout

PCB123

13626 S Freeman Rd. Mulino, OR 97042 USA
Phone: (800) 228-8198 x236 / Fax: (503) 829-6657

DOWNLOAD THE NEW VERSION @ www.pcb123.com

Have design files?

YOUR TARGET FOR PROTOTYPE CIRCUIT BOARDS

LEADING INTERNET SUPPLIER

pcbexpress

PCBexpress.com

PROTOTYPE CIRCUIT BOARDS

Successfully selling online since 1997.
Experience you can trust! Try us out today at:
www.PCBexpress.com/servo

Enter the link for a special SERVO discount!

Circle #80 on the Reader Service Card.

TABLE OF CONTENTS

Departments

6 Publisher's Info
7 Bio-Feedback
8 Tetsujin Info
38 *SERVO* Bookstore
47 Robotics Showcase
70 New Products
71 Hack-A-Sapien
78 Events Calendar
82 Advertiser's Index

Columns

6 Mind/Iron
14 GeerHead
23 Robytes
28 Brain Matrix
30 Rubberbands
52 Menagerie
58 Ask Mr. Roboto
62 Tetsujin Tech
72 Robotics Resources
80 Appetizer

Take A Sneak Peek!

DARPA Grand Challenge

Coming 10.2004

On the Cover ...

Tetsujin competitors Jascha and Scott Little share a 3D model of their bionic exosuit. More info on this amazing event is on page 8.

SERVO Magazine **(ISSN 1546-0592/CDN Pub Agree#40702530)** is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. APPLICATION TO MAIL AT PERIODICALS POSTAGE RATE IS PENDING AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, 430 Princeland Court, Corona, CA 92879-1300** or Station A, P.O. Box 54, Windsor ON N9A 6J5. cpcreturns@servomagazine.com

SERVO 09.2004
VOL. 2 NO. 9

Features & Projects

9 In the Shop with FIRST Team 1079
by Bryce Woolley

18 Putting SMAs to Work
by Don Wilcher

26 Return of the Mechanical Man
by Tom Carroll

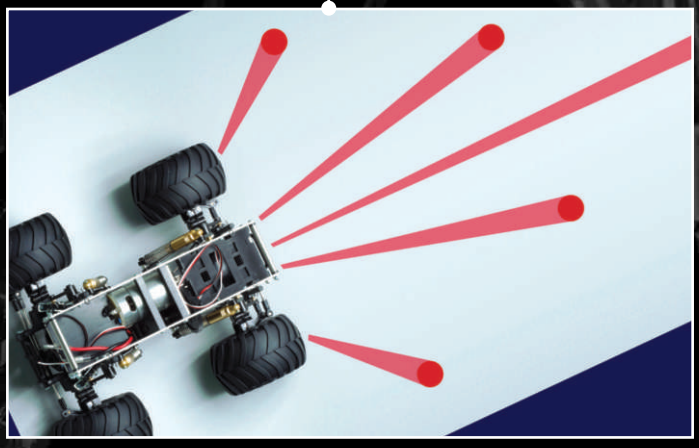
34 Trigonometry Tips for Small Processors
by Bill Brew

41 Pimp My Tank
by D. Jay Newman

48 BEAM Neurons and Sensors
by Thomas Gray and J. Wolfgang Goerlich

53 Following Doc Octobot
by Roger G. Gilbertson

63 Sensor Fusion for Robo-Magellan
by Michael Miller



Mind / Iron

by Dan Danknick

In the 1970s, the term "glass ceiling" became popular with female executives that found their career advancement mysteriously impeded — they could see the next rung of the management ladder, but they couldn't seem to reach it. The cause was unfair expectations of male co-workers, not the abilities of the females.

I often find myself wondering if the robotics industry is trapped by a glass ceiling of sorts. It only takes a short conversation with a robot innovator or developer to make me ask, "Why don't we have robots integrated into society already?" In her review of the movie *I, Robot* (page 80), Dr. Joanne Pransky touches on the complexities of a human society full of robots. But to me, that seems like a future still light years away. Why? I think it comes down to one of two reasons: either researchers are not working on the right things, or humans aren't interested in a robotic future.

Consider the first, that the products of design labs around the world are not meeting the needs of society. Robot development is a business and, as such, needs to meet certain economic criteria to remain viable (identify a market, sell at a price that turns a profit, etc.). This tends to push robotics behind the scenes to unglamorous jobs that not only are uninteresting to people, but largely unrecognized.

Take intelligent fasteners, as an example. Never heard of them? I hadn't either, until I read a paper by Dr. Seshu Seshasi of Textron Fastening Systems. Consider a special bolt that has a microchip, shape memory alloy core, and a serial interface. Give it power and an (encrypted) command, and it responds by heating the SMA to release whatever it fastens. Far fetched? Not really, as Textron is developing these to prevent the theft of airbags, music players, and spare tires in automobiles. An intelligent fastener is about the most un-sexy robotic application I can think of. But, Textron wouldn't be working on them if there wasn't a market (back to that economy thing again).

Projects like these sprout like weeds through labs around the world; I'd run out of space listing them here. So clearly, the problem isn't that researchers are working on the wrong things. What about society's view of robots, then?

The world has come a long way in the past 100 years. We can talk to people across the country with the push of a button, our dentists use lasers to fix our teeth, and we analyze terabytes of data looking for radio signals from alien life. But are we ready to let a Nestor NS4 walk the dog? Yes, I think we are, but that isn't how it is going to happen — it is certainly "cool," but it isn't practical from a business sense.

Robots will continue to solve the problems that deal with manufacturing efficiency, safety, and security. Because that is where the money is. We won't wake up one day to a robot cooking breakfast for us, but we will most certainly wake up in a temperature controlled house, a kitchen that reorders food automatically, and a navigation system in our car that gets us to work safely and on time. Robots solve the problems that we don't want to attend to anymore.

So what of this glass ceiling? Looks like my premise was inaccurate. For those of us in robotics, the sky is the limit! SV

Published Monthly By
The TechTrax Group — A Division Of
T & L Publications, Inc.
430 Princeland Court
Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
www.servomagazine.com

Subscription Order ONLY Line
1-800-783-4624

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
robin@servomagazine.com

MANAGING/TECHNICAL EDITOR
Dan Danknick
dan@servomagazine.com

ASSOCIATE EDITOR
Alexandra Lindstrom
alexa@servomagazine.com

CIRCULATION DIRECTOR
Mary Descaro
subscribe@servomagazine.com

WEB CONTENT/STORE
Michael Kaudze
michael@servomagazine.com

PRODUCTION/GRAPHICS
Shannon Lemieux

STAFF
Janessa Emond
Kristan Rutz

OUR PET ROBOTS
Guido
Mifune

Copyright 2004 by
T & L Publications, Inc.
All Rights Reserved

All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *SERVO Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *SERVO*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *SERVO*. Please send all subscription orders, correspondence, UPS, overnight mail, and artwork to: **430 Princeland Court, Corona, CA 92879.**

BIO->FEEDBACK

Dear *SERVO*,

In a recent issue of your magazine, you had an article on CNC machines, mills, and such. You mentioned that CNC machines start at $1,000.00 to $1,500.00. While commercial units may cost that much, you can build your own machine for much less. I am currently working on one such machine. You can buy plans for CNC machines (among other things) from a man named John C. Kleinbauer, for under $40.00. The machines are made with simple shop tools (scroll saw, drill press, etc) and are made of simple materials (plastic, parts from hardware stores, other easy-to-get items), too. Best of all, he runs an online discussion board for people who have bought his plans to get help and exchange ideas. You can find his website at **www.crankorgan.com** For some time and effort, you can make a machine for well under $1,000.00, even if you have to buy all the tools to make it with.

Michael Cook
via Internet

Dear *SERVO*,

In the piece "A Synthetic Nervous System For Robot Locomotion," there seems to be a discontinuity in the text of the article as it crosses from page 19 to page 20: "... more interesting wave forms can be obtained by frequency lampreys — one might try op-amp phase shifters ..."

This makes no sense. Also, there is no mention of Circuit 4 in the article. I suspect that Circuit 4 has been placed over the text. Any chance of getting the missing text please?

Brian Bibby
via Internet

The correct text should have been:

"By phase locking two of these circuits together, as in Circuit 2, more complex and interesting wave forms can be obtained by frequency modulating one or both circuits to change its phase orbits and see the effects on your robot and how it walks. Circuit 4 shows a simple means of 180° phase modulation. Q11 acts as an inverter so that one signal can control Q12 and Q13, allowing the oscillators to have variable phase coupling. On your oscilloscope, this allows you to control the rotation of the circle that controls forward and backward motion in the robot. For some robots — like snakes and lampreys — one might try op-amp phase shifters (say 10 or 20°) to allow for smoother undulation."

Apologies for the omission, we're still trying to figure out how this happened in production.

— Editor Dan

Dear *SERVO*,

In Figure 12 of my August "Rubberbands and Bailing Wire" column, the comma in 600,000 is incorrectly shaded blue since 0{2-4} won't match the comma. Thanks!

Jack Buffington
Columnist

Dear *SERVO*,

Thanks for a great magazine. It reminds me of the good old days when *Byte Magazine* had good technical content, and *Doctor Dobb's Journal* was still concerned about BASIC interpreters that fit into 4K.

Gavin Thomas Nicol
via Internet

Dear *SERVO*,

Mr. Nicholas Blye's interview with Mr. Mark Tilden and his technical explanation of the biomorphic biped robot named RoboSapien was outstanding. It alone changed my mind about canceling my subscription. In fact, I can say that article alone was worth the subscription price. I have renewed my subscription and as long as I see work like that, you will have a dedicated subscriber.

Edward DeMayo
New Fairfield, CT

Hobbyists, Designers, Students:

Why wire-wrap when you can get Prototype PCBs for much less?

Our Standard PCBs include Tooling, Mask, and Legend

2 Layer 5 days $13 each

4 Layer 5 days $33 each

Our "Bare PCBs" have no mask and no legend but can give you even more SAVINGS!

$40 lot + 60 cents per square inch for 2 layer Bare PCBs

Easy Online Ordering at:
www.PCBFABExpress.com

PCB FABEXPRESS
High Quality PCBs @ Low Impact Prices

Email: support@pcbfabexpress.com Tel: (408) 857 0039

Circle #115 on the Reader Service Card.

A glimpse into the future of robotics!

When we announced Tetsujin 2004 back in March, we knew it would raise some eyebrows. The concept — a live competition where people step into strength augmentation suits to lift weights — was groundbreaking. As expected, it has attracted the attention of the best and brightest fabricators and engineers from around the country. Move over, DARPA Grand Challenge!

Next month is your chance to see what these seven amazing teams have come up with! Join the top minds in industry, as well as the worldwide media at RoboNexus — the nation's largest event for emerging robotics technologies. Online registration is simple and quick — don't delay!

If you are a decision maker for a company that wants unique and positive exposure, the exosuits of Tetsujin 2004 may be the channel you've been looking for. To explore advertising and marketing opportunities further, drop an Email to **tetsujin@servomagazine.com** or call (951) 371-8497 and we'll pair you up with a team that embodies the spirit of innovation, as well as the positive discipline of a "can do" attitude.

RoboNexus

The International Business Development, Educational and Consumer Event For Personal, Service and Mobile Robotics

Santa Clara Convention Center, October 21-23, 2004

www.robonexus.com

FIRST FRENZY 2004

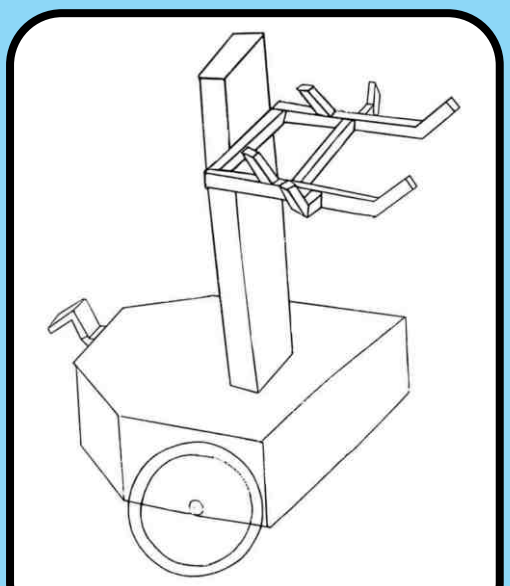
BY BRYCE WOOLLEY

What is FIRST?

FIRST is an organization founded by the famous inventor Dean Kamen, the man responsible for such innovations as the Segway Human Transporter and IBOT Mobility System (the wheelchair that can climb stairs). FIRST is centered on a high school robotics competition where teams from all over the United States — and even from Canada, the UK, Brazil, and Puerto Rico — build robots to compete in a new game every year. FIRST stands for: For Inspiration and Recognition of Science and Technology. It seeks to inspire young people to pursue careers in those fields. The Mission of FIRST is to "design accessible, innovative programs to build self-confidence, knowledge, and life skills while motivating young people to pursue opportunities in science, technology, and engineering."

For those readers who are not associated with a FIRST team, this article will provide a window into a bustling robotics community that you probably never knew about. For the readers who are associated with a FIRST team, this article will provide insight into the dynamics of another team during this year's build. Hopefully, this will inspire the former to look into the local teams and become involved and provide a frame

FIRST FRENZY 2004



SKETCH #1

of reference for the latter, by which they can appreciate their own team.

I am Bryce Woolley, currently a senior in high school and co-captain of FIRST team #1079, CREATE. We hail from Chaparral High School in Temecula, CA. CREATE stands for Chaparral Robotic Engineers And Techno Explorers. As of 2004, Team #1079 is an average team in most respects. We have a relatively small group of about 12 students and several engineering mentors, who are generally the fathers of team members — my father is one of them.

Our team has money issues and limited corporate sponsorship (like most teams). We also lack the significant professional engineering help that many veteran teams enjoy. Our school support is also weak, although not completely lacking. Thankfully, we have a supportive community around us; many local businesses and machine shops help us out.

Don't get the wrong idea that every FIRST team is like us; we are only an average young team. There are some teams with hundreds of students and several enthusiastic teachers involved, tens of thousands of dollars in sponsorship, and an official school course centered on FIRST robotics. We, on the other hand, are frequently in debt, short-handed, and build our team's robot in our garage on weekends, but we have a blast doing it just the same. Now, this is the story of our team's experience with FIRST Frenzy 2004.

The Kick-Off

After we register our team in the fall and pay the required fees by December, we are ready for the kick-off event in January. This year, the kick-off was on January 10. A small portion of the team traveled up to the University of Southern California campus for the LA area kick-off event, where we would find out the game for this year and get our kit of parts.

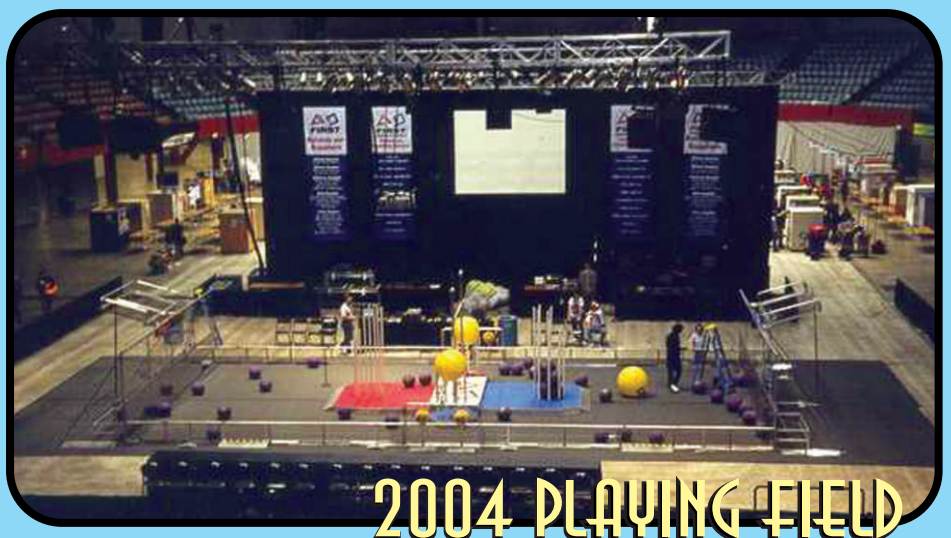
"Kit of parts?" you may ask. Well, the way the competition works is that every team gets the exact same kit of parts to build their robot on the exact same day at various remote kick-off events. The kit contains all of the necessary components to build the robot, like the control system, motors, wheels, etc. Even though all of the parts are the same, it is truly amazing to see the variety of designs at the competitions.

After each team receives the kit of parts, they only have six weeks to design and build a functioning robot. Other restrictions include size and weight limitations (36" x 30" x 60" and 130 lbs). Wow, it's just like the real world, with tight schedules and seemingly impossible deadlines! At USC, everyone crammed into a small room to watch the satellite feed from the East Coast, where Dean Kamen, Woodie Flowers (Pappalardo Professor of Mechanical Engineering at MIT), and Dave Lavery (Program Executive for Solar System Exploration at NASA) announced the new game for this year. The game was called "FIRST Frenzy" — an eclectic mix of elements from previous games and some new aspects, too.

A quick reference to the picture of the playing field will greatly aid in your understanding of this complicated game. One of the most important parts of the game is the robot/human player relationship. The robot cannot score points with the balls, only the human player can. The human players cannot score points unless the robots pass the balls to them. In short, the points are scored in the game by the human player by throwing the small balls into either the stationary goal or the mobile goal of their alliance color (red or blue).

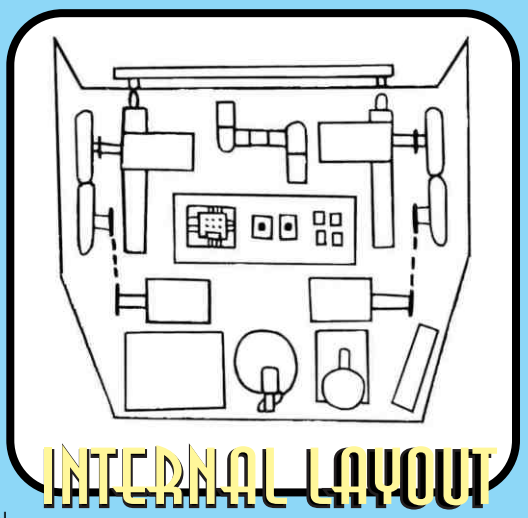
If the robot can hang on the 10-foot high pull-up bar in the center of the field, the team scores an additional 50 points. The points of the smaller balls in the goals can be doubled by having the robot place the big, yellow, 2x-multiplier ball on top of the goal. Four robots are on the field at once, paired in teams of two each for the red or blue alliance.

After hearing the description of

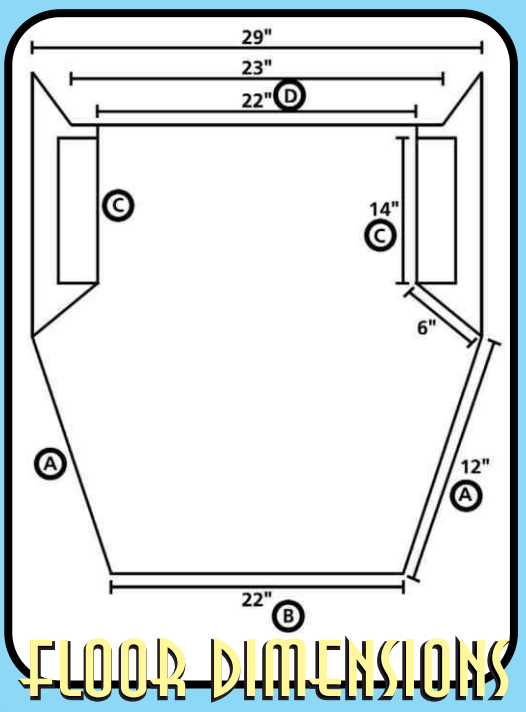


2004 PLAYING FIELD

the game, our team gathered together to brainstorm the design and functions of our robot.



INTERNAL LAYOUT



FLOOR DIMENSIONS

Brainstorming

The tough part about brainstorming is reconciling everyone's unique ideas in order to agree on one functional design. First of all, we all had to decide which tasks to accomplish and how to do so. We gathered all the team members into a circle and each person stated his or her opinion on what he or she thought would be the most important task to complete and any design ideas and what he or she would like to focus on.

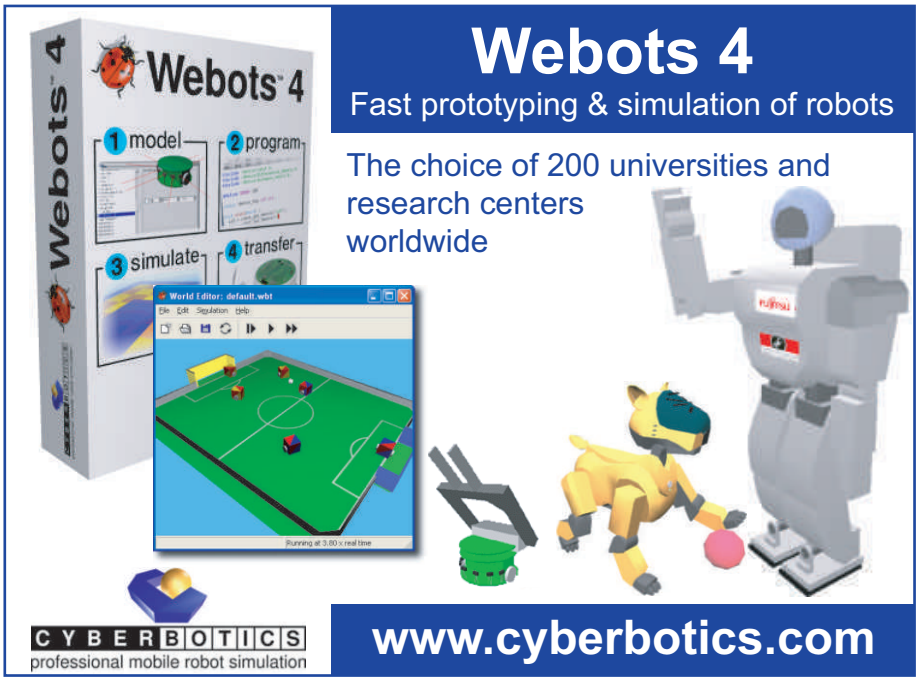
The team unanimously agreed that our robot should be fast and maneuverable. This would allow us to control the balls and mobile goals with ease. The team also decided that the robot should be able to handle the small balls with accuracy. This would best be accomplished through angled extensions at the front of the robot, as well as a pneumatic kicker to push the balls into the chute to the human player. We also saw mobile goal manipulation as a key factor for victory.

If we could control the mobile goals, we could move ours so we could score more easily or move our opponent's mobile goal so that they could *not* score with it at all. The shape of the front of the robot — with the angled extensions — conformed nicely to the shape of the goal. That ensured that it would be easy to move into pushing position and/or close enough for manipulating the large multiplier ball — an important consideration when you're driving by remote control.

In the back, we decided that a grabbing mechanism for the goals would also be helpful; the grabber would latch onto the lip of the goal's base and we could drag it to an advantageous position. Our team also focused on what would score the most points and we naturally fixated on the big yellow multiplier ball.

Several of the younger team members had recently taken an "Arms and Lifts" class at a SCRRF (Southern California Regional Robotics Forum), sponsored robotics workshop at CSUN (California State University, Northridge). They were eager to apply their new knowledge and come up with unique designs. In their enthusiasm, they began to formulate grandiose schemes, where the arm could retract, extend, clamp with pneumatic fingers, hold the ball, cap the goals with the ball, and even extend and attach to the pull-up bar and allow the robot to hang. The more realistic members of the team — who were better acquainted with the tight budget and our lack of sophisticated arm-building technology — did not want to disenfranchise the younger members by immediately modifying their ideas, so they merely introduced their own.

Thus began the "great arm debate" that was destined to witness



Circle #73 on the Reader Service Card.

many a team gathering, committee meeting, tribal council, and the like before it was finally resolved. The ultimate solution was a simplistic, yet functional, design that worked much like a forklift to handle the multiplier ball. (The arm design will be more thoroughly discussed later.)

One of the most critical parts of the robot is the floor. The floor dictates the rest of the robot design and is a major factor in the robot's overall strength. The floor also determines the robot's shape. One of the most important principles that went into the design of our floor was CFE — Cool Factor Engineering. This probably comes from our background in combat robotics, where the intention is entertainment. Even though FIRST is not necessarily for entertainment, we take pride in having robots that look good.

CFE also distinguishes our team's robots from others because ours are always very polished and actually sport design features that are purely aesthetic and non-functional. Thus, our floor was designed in a trapezoidal fashion instead of as a conventional, uncool, square. With the angled extensions on the front, too, the floor was looking quite interesting.

With the floor design agreed upon, we moved on to the wheels. The kit came with two large wheels that we were initially drawn to; they would be able to climb up the small stairs in the middle of the arena with ease. This idea was functional, but the only problem was that it was not cool. All of that changed with a visit from one of our esteemed professional robotics mentors.

He liked our idea of having a freewheeling caster wheel in the back with the driven wheels in front, but he suggested that we replace each single big wheel with a pair of smaller wheels. This idea was indeed cool. We eventually decided that, on each side, the two wheels would be driven by two different, independent motors matched as closely as possible through sprocket

Circle #105 on the Reader Service Card.

reductions. The caster wheel also conformed to our CFE specifications because it would be a custom design made out of two wheels.

The Build

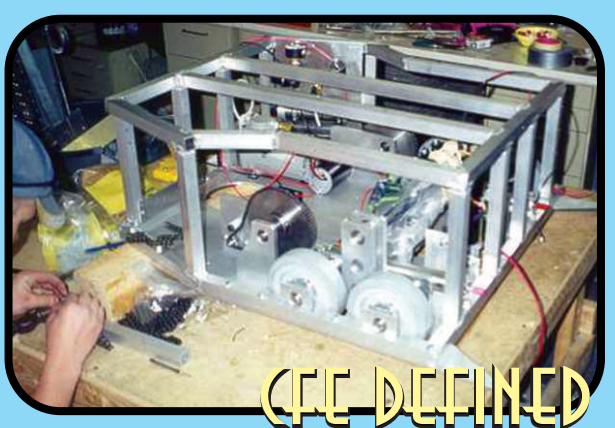
As our team entered the build phase of the project, we were eager to institute some procedural changes according to what we had learned the previous year. As last year was our rookie year, we did not have any plan for administering the build, aside from "everybody work." This year, we would adopt a trick used by many of the veteran teams; we would delegate the systems of the robot into separate projects. An experienced member of the team would head up each project. I would head up the pneumatic kicker system and goal grabbing system.

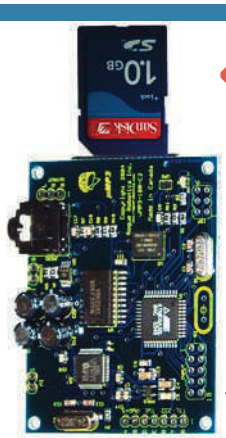
Other systems that were delegated included work on the drive train and frame. Many of the younger team members were eager to take on the challenging arm system. They wanted to build wooden prototypes to figure out the most efficient design, which proved quite useful. Even after the arm concept was "finalized," it underwent a series of design changes before it arrived at its functional and effective state (those changes and details will be discussed next issue).

This delegation of jobs promoted efficiency and teamwork among the group — two precious and elusive commodities during the long haul known as the FIRST build. However, this harmony was unstable at times. As all FIRST teams know, these builds can test the limits of friendship, patience, and even sanity. Our team is no exception, no matter how hard we try to conform to the rules of "gracious professionalism."

Things started to go awry with the great music controversy. Some of the adults on the team insisted on the folksy tunes of the Kingston Trio while some of the students on the team agitated for the more aggressive music of Metallica. As if that was not enough, the childish antics of some of the team members were flirting with disaster as the DADCON started to go up. (DADCON is the condition of the dad; it goes up when any dad present at the build gets stressed).

Will the robot get done in time? How will the arm turn out? Will the team choose the Kingston Trio or convince the dads to listen to Metallica? Find out the answers to these burning questions and more about Team #1079's experience with FIRST Frenzy in the next issue of *SERVO Magazine*! SV



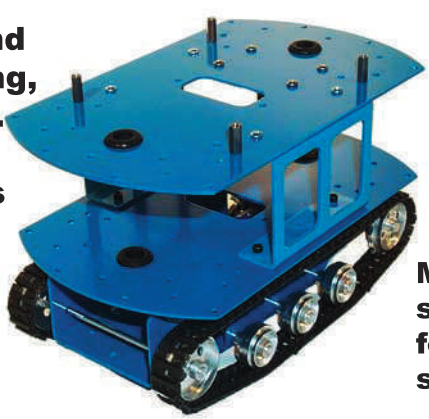


uMP3 *Now Shipping*
Serial MP3 playback module

Great for sound F/X, advertising, robot speech..

SD/MMC cards w/FAT16/32

Rogue ATR
expandable tracked robot system



New from ROGUE robotics

www.roguerobotics.com
(866) 99 ROGUE Fax (416) 238-7054

uMMC
MMC/SD card serial interface for FAT data storage

Circle #49 on the Reader Service Card.

GEARHEAD

by David Geer
geercom@alltel.net

Mark Pauline and Survival Research Labs Present

"The Most Dangerous Shows on Earth"

This Eldorado board blaster can shoot its wooden ammo 1,500 feet into the air!

It Takes a Dangerous Man to Create a Dangerous Show

Long ago and far away (for those who don't live in California, anyway), it would appear to some that poor Mark Pauline lost his mind. Twenty-five or so years later, he hasn't recovered — from his deliriously dangerous passion, that is.

You see, those many years ago, Mark decided that it would be a good idea to develop spectacles of entertainment that involve lethal, multi-thousand pound robots and mechanizations. That not being enough, his audiences really needed to sit close enough to the show to risk injury (though no one has been hurt to date, the object being entertainment, not elimination).

Finally, it was — of course — necessary to have the occasional fracturing of the local, state, and federal laws that still got in the way after getting a variance. Mark naturally came upon a nice, malapropos name for the organization that embodied these death defying theatricals; he named it Survival Research Labs.

Research That Would Make Frankenstein Proud

Moving quickly from our intro and through the shortest disclaimer on record (insert subliminal message that Mark is not really crazy here), we arrive

at the gory details.

Survival Research Labs creates major theatrical spectacles (as major as a venue supporting 2,000-3,000 audience members can get) where all of the performers are robots, weighing from 1,000 to 6,000 lbs each. (Okay, so I missed this one for "Mammoths and Dinkies 2." I'm only human — not a single Borg part in my body.)

From the show "DELUSIONS OF EXPEDIENCY", premiering on *10 Years of Robotic Mayhem* DVD.

There is a set or stage as with any performance of actors, as well as a script of roles to be played. As with any good TV show or movie, the audience is captivated, drawn in, escaping reality and the world that encases it.

Obstacles

The problems with building the robots are as formidable as the robots are themselves, but not insurmountable, since Mark's day job more than enables him to fund all of SRL's machining needs. The biggest problems are being able to get things done on schedule and negotiating the safety issues that surround fabricating the robots.

The ideas come easy to Mark. Mark Pauline writes all the scripts for each show. SRL performances are never advertised, creating a sufficient draw by word of mouth. Mark pays for them out of his own pocket, never commercializing his effort.

Danger! Danger!

Mark's robotics include a pitching machine, basically a 500 cubic inch Ford Eldorado motor that shoots 2 x 4s like a gun by using two spinning truck tires that rotate at 200 mph in opposite directions. The six-foot 2 x 4 bullets are loaded into the gap between the two tires at a rate of two-boards-per-second.

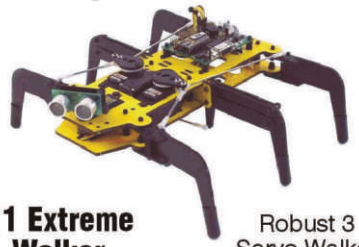
The boards are immediately fired from between the tires at that same 200 mph. This Eldorado board blaster can shoot its wooden ammo 1,500 feet into the air. (To think poor Mr. R. J. Gattling wasted all that time on his comparably feeble pea shooter. Hmmm.)

As part of some performances (each unique from the rest), this device has been used to destroy structures the size of small buildings in six seconds flat. Mark calls his enormous 2 x 4 "'pistola" the most dangerous robot in the world. Believe me, I didn't stop to argue.

Each performance is intended to communicate more abstractly than concretely and employs 20 to 30

The Eggman explodes while the Inchworm looks on.

www.lynxmotion.com

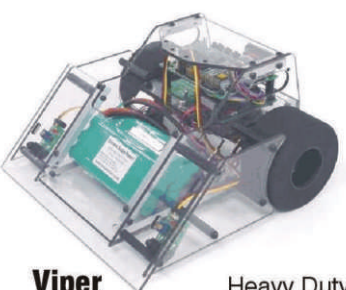


H1 Extreme Walker — Robust 3 Servo Walker

H2 Extreme Walker — Performance 12 Servo Walker

Lynx 6 Robotic Arm — Five DOF Arm

4WD2 Rover — Articulating Chassis

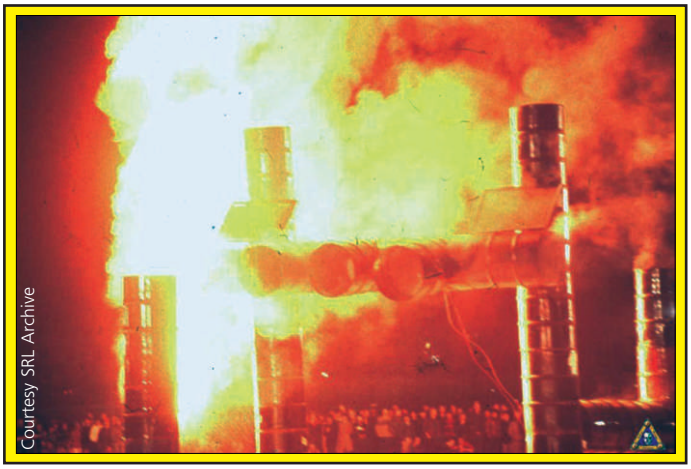


Viper Sumo Robot — Heavy Duty Rugged Design

We have many more unique robot kits. Our robots feature:

- Precision Laser-Cut Lexan
- Preassembled Electronics
- Custom Aluminum Components
- Injection Molded Components
- Very High Coolness Factor

Toll Free: 866-512-1024
Web: www.lynxmotion.com



From the show "DELUSIONS OF EXPEDIENCY" premiering in the *10 Years of Robotic Mayhem* DVD.

Director Jon Reiss lines up a shot on the set of "A Bitter Message of Hopeless Grief."

robots of all sizes, all running simultaneously for a show of about an hour's duration. Each robot is as unique as the shows are from one another, yet each approaches the same level of threat that it can pose. Suddenly, "The Most Dangerous Show on Earth" doesn't seem so out of place as a description.

Of Vices and Venues

Mark insists that venues allow people to sit as close as possible to these performances. In most cases — when the shows are permitted — he gets his way. In many instances, the shows have been banned. In fact, Mark believes they are the most banned troupe of artists in the world.

In addition to the violence, Mark describes the shows as having a vicious satirical bent. Mark makes it crystal clear that his shows are not family entertainment. All this makes it more difficult to get approval to do the shows. Mark works with institutions like museums or gets people on board at the mayoral level to get the clout he needs to drive the performances forward.

Mark Pauline and SRL have done 56 different shows since 1979. Each one has been as unique as a snowflake (though not necessarily as pretty), using a different array of machines — old and newly fashioned, each one dressed differently than before, and maneuvered or employed in a way particular to the theme of the given show.

The shows are Mark's "calculated hallucinations," intended to stay with the audience members throughout their lives. Each show's theme is

A Little Background

Creator and founder Mark Pauline began pouring his heart into his art and his machines in November of 1978 in the form of Survival Research Laboratories (SRL). An organized team of highly trained, experienced, and creative engineers and technicians, SRL is devoted to an art form that they describe as, "redirecting the techniques, tools, and tenets of industry, science, and the military away from their typical manifestations in practicality, product, or warfare."

Fifty-six performances, soon to be 57, have been put on display across the US and Europe to demonstrate to the world exactly what that means. Anyone who has witnessed these events is very clear on that meaning.

Robots, machines, and FX combine to portray satirical themes that are harsh and abrupt, at the very least. Whatever the particular theme, it is branded on the minds and hearts of the audience members — and the technicians "pulling the strings" of these robotic puppets — for life.

Jon Reiss and Mark Pauline on set.

The Doom Show

Around 2000, as the previous millennium was near a close, Mark Pauline and his creation, Survival Research Labs, brought forth "The Doom Show," an end of the millennium show that Mark describes as, "a calculated forecast of ultimate doom."

The show included an eight-foot tall, 35-foot wide countdown clock that counted down to zero and back up again at random. The clock was destroyed and burned during the show.

expressed by offering up "visual puns"' and "sight gags."

Stories of True Crime

Mark mentioned quite calmly that SRL always tries to commit serious criminal violations at every event, with a special interest in breaking federal laws. Here's an example: For a New York City show, SRL made $3 million in perfect, three-color, counterfeit money, which he described as, "very passable money."

SRL used a catapult to fire bombs up into the air; they were actually packages of an eighth to a quarter of a million each of the bogus bills. (Mark assures me that the statute of limitations has run out on this particular offense.) The packages were fired 100 feet up and, upon detonating, they dropped all this cash to the ground and onto the audience. The audience and even the New York City police officers present were grabbing it up. SRL actually owns some videos of these New York City cops stuffing the faux dough into their pockets.

As if that weren't hilarious enough, there happened to be a carnival going on next door to that performance. The very next morning — while the SRL gang was still cleaning up the debris — they were set upon by a mob of carnies who had gotten wind of the show and all this fake money.

The carnies surrounded the place and started picking through the site, hoarding all the money that was left. They actually took it back to the carnival to use it as change for their patrons. People were passing it all over New York City for weeks; yet, surprisingly, nothing ever came of it. The SRL crew waited, expecting a visit from federal officers, but it never came.

The Shockwave Cannon from the "Deliberately False Statements ..." show.

Pipe Dreams

At a recent show, SRL employed an eight-foot styrofoam teddy bear. The show was at a modern art museum in Berkeley, CA. Big black pipes support the whole building. SRL built a fake black tube and set it up to look like one of the supports. Somehow, the giant teddy bear was sitting atop this phony support. SRL had a huge machine come and rip that pipe apart — as if it was ripping it out of the building. Then it tore the teddy bear apart.

Coming Events

SRL just completed building 21 crawling robots — each one about nine feet tall. These will be used in the next show, which will be held in December in the Los Angeles, CA area. R/C operates the robots. They look like crawling soldiers, so SRL calls them sneaky soldiers.

Because so many soldiers have been getting killed lately, SRL is going to put on a show where they bring the dead soldiers back to life in order to kill them again. They will have the robot soldiers crawling all around while other machines are destroying them.

Ultimately, the art world will consider itself fortunate to have the achievements of SRL as a part of its history, even if they do not do so today. SV

DVD Release — "Mayhem" is 10 Years of the Most Dangerous Show on Earth

It's now the 25th anniversary of the birth of Survival Research Labs. SRL, one of the earliest — if not the earliest — source of gigantic, awe-inspiring robotic spectacles, began with a dream, er, nightmare of a dedicated team and a couple bots. Today, between dozens and perhaps hundreds of machines have been retooled to create shows that do, indeed, rival anything Barnum and Bailey put forth.

These 25 years are celebrated with 10 of them engraved in the DVD *10 Years of Robotic Mayhem* (check with SRL or Jon Reiss for availability).

The DVD includes films like "A Bitter Message of Hopeless Grief," which appeared in film festivals worldwide, including the Sundance Film Festival, Berlin, and some 30-odd others. Also included is "The Will to Provoke," a documentary of SRL's first European tour.

The DVD comes complete with parts of four other SRL documentaries, comprised of footage from early performances, and the first show where audiences were up close and personal with the mechanized menagerie.

Resources

Buy the SRL DVD from
www.parasiteproductions.com/roboticmayhem

The Internet home base for Survival Research Labs is at
www.srl.org/

Jon Reiss, who has immortalized SRL's work on DVD
www.jonreiss.com/

An SRL home page, but not *the* SRL home page, is at
www.canuck.com/Srl/srl.html

EXPERIMENTING WITH SHAPE MEMORY ALLOYS

Build a NanoMuscle Actuator Switch

Part 2 — by Don Wilcher

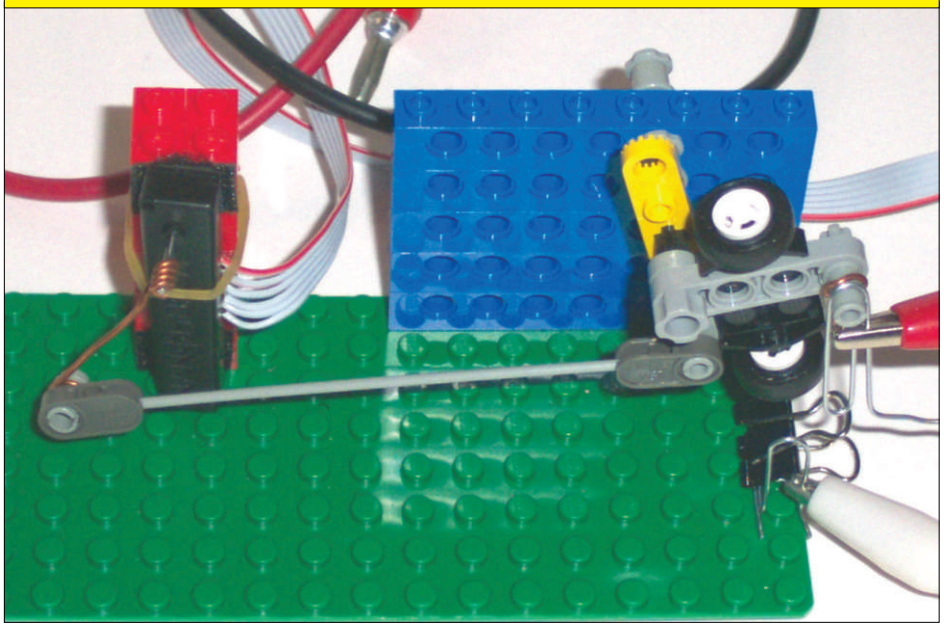
In last month's article, the basics of Nitinol wire were discussed along with a brief history of Shape Memory Wire. The NanoMuscle Actuator's internal control circuit architecture was explained, along with a simple electronic controller for experimenting with the SMA device.

To further explore this small, MEMs-based motor, a prototype actuator switch built from LEGO bricks will be discussed in this installment. Also, a design discussion on wiring the NMA-based switch to a small DC motor will be presented.

Building the NMA Switch

The NMA switch is quite crude in its physical appearance, but most experimental prototypes usually are. The NMA switch is built using LEGO bricks as the foundation for establishing a framework for the actual electrical contacts. The electrical contacts are made with alligator test leads, a binder clip, and a paper clip. Figure 1 illustrates the finished NMA switch and Figure 2 shows the parts breakdown of the electrical component.

Figure 1. The complete LEGO-based NMA Switch.



The use of LEGO building bricks in the construction of the NMA switch was selected because of the ease of design modification of a mountable structure for the nanomuscle rotary motor.

Also, no special tooling is required, so the build time of the electromechanical switch is relatively quick. An erector set can also be used as a mountable structure for the NMA, as well.

Special precaution should be taken when using an erector set along with the NMA device because of the metal surfaces. The ribbon cable should be routed away from the erector set's metal edges to eliminate the possibility of a short circuit when energizing the NMA rotary motor. The interlocking bricks allow for various mounting structure configurations.

The modular stacking design of the 2 x 2 bricks shown in Figure 3 was based on the requirement that the NMA rotary motor have a reciprocating motion translating into a linear swing movement of the "paper clip" contact of the switch. Figure 4 shows a free body diagram for the required paper clip motion.

Basically, the "rotating" paper clip contact will touch the "fixed" binder

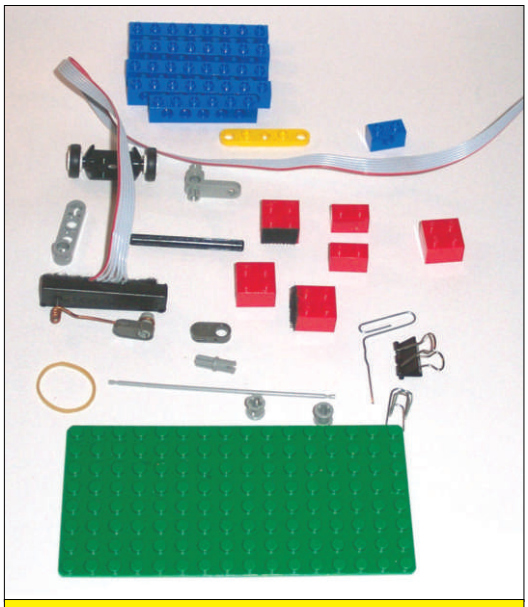


Figure 2. Parts breakdown of the LEGO-based NMA Switch.

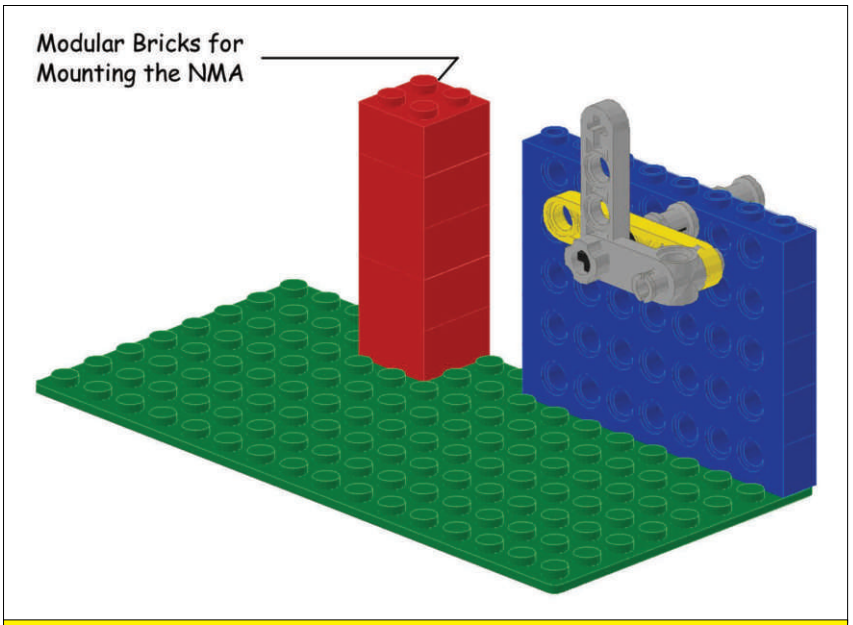


Figure 3. Modular stacking design for NMA mounting.

clip, thereby allowing electrical connectivity to occur at the mechanical interface. The NMA rotary motor is the prime mover for the electromechanical switching system.

Once the mounting structure design is selected and built, move the rotating parts with your hand to make sure there is no mechanical binding. If all goes well with this quick test, the NMA switch can be actuated with the simple electronic controller circuit discussed in Part 1 of this project.

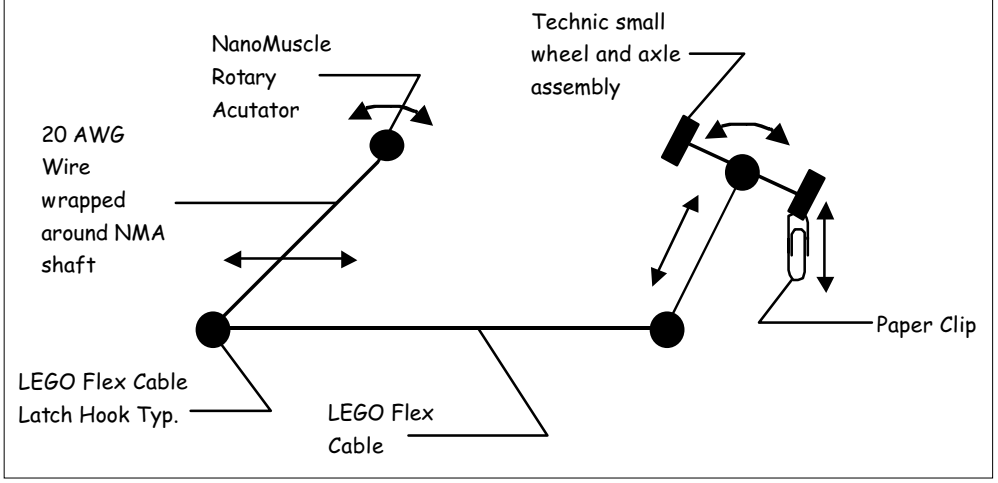
Testing the NMA Switch

Testing electronic components requires a systematic plan for checking the I/O (input/output) of the device under test (DUT). Having a block diagram of the DUT available helps in testing it because the electrical names of the I/O are known. With this information present, the appropriate test equipment can be selected for checking the DUT. If a manufacturer's block diagram of the DUT is not available, one can be created quite easily.

By drawing a box that represents the DUT, all of the electrical inputs can be shown using lines with attached arrows placed on the left side of the object. The arrow head(s) should be pointing toward the box to show signals going into the DUT.

Next, lines with arrows going away from the box can be added to the object representing outputs. Like the input signals, the outputs should be defined, as well. Figure 5 shows the NMA Switch block diagram. In the diagram, the electronic controller circuit signal is the input, the output is the NanoMuscle rotary motor, as

Figure 4. Paper clip free body diagram.



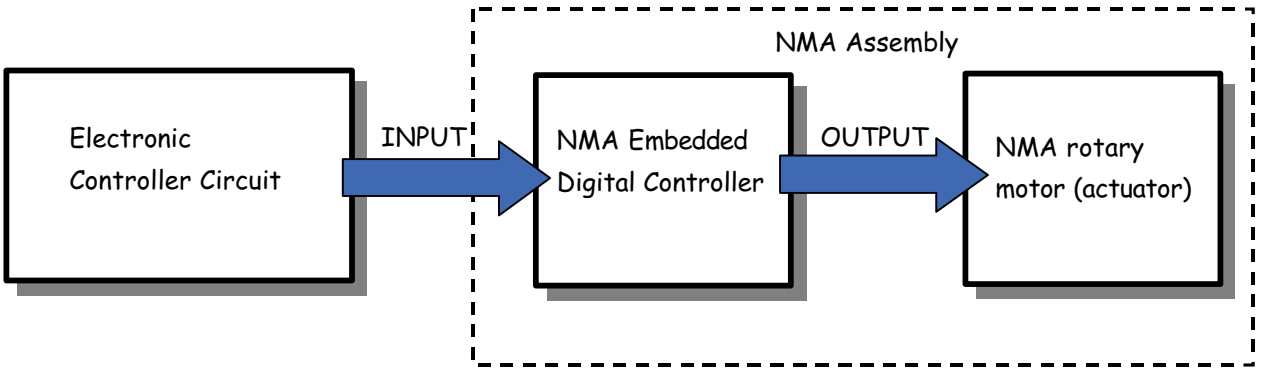


Figure 5. NMA switch block diagram.

Parts List

LEGO Parts — NMA Mounting Structure

1 Green	Baseplate 8 x 16
6 Red	Brick 2 x 2
1 Black	Technic axle 5
6 Blue	Technic brick 1 x 8 with holes
2 Light-Gray	Technic bush
1 Light-Gray	Technic liftarm 1 x 4
1 Yellow	Technic liftarm 1 x 5
1 Light-Gray	Technic steering arm
1 Light-Gray	Technic small wheel and axle assembly
2 Brown	Technic flex cable latch hooks
1 Gray	Technic flex cable No. 10

Electronics — Controller circuit

5.1K resistor, 1/4W
6.67Ω resistor, 1/4W
60Ω resistor, 1/4W
NanoMuscle rotary actuator (motor) RS-125-CE
Electromechanical switch
Solderless breadboard
+5 VDC power supply

Miscellaneous

Wires
DMM (Digital Multimeter)

Note: It is assumed the electronic controller circuit discussed in last month's article is already assembled. If not, refer to Figure 7, and use the electronics parts list to construct one. After building the circuit, use the test procedure in the article to check out the NMA switch.

well as the processor of the DUT system.

The next step in testing the NMA Switch is to develop a test procedure that can be used to check the electrical function of the nano-based electric switcher.

A state machine is a graphical tool that can be used to develop a testing procedure for the NMA switch. Basically, a state machine is a concept map capturing function and feature knowledge about the target product's operating attributes.

To draw a state machine, a set of symbols is used to capture the states and transitions of the device behavior or function. Circles represent states with arrows defining the transitions between them, due to some event. The state diagram starts with the DUT turned off.

Next, a transition from "Power Off" to the "NMA Rotary Motor Off" state is accomplished by applying power to the electronic controller circuit.

With the electronic circuit energized, pressing the electric switch of the electronic controller circuit will cause the nanomuscle rotary motor to turn on and stay on until the electric switch is released.

Figure 6 shows the complete state

machine for the NMA switch. The schematic circuit for the electronic controller circuit shown in Figure 7 provides an additional resource to create the state machine.

With the state machine completed, the test procedure can be written to validate the NMA switch. In writing the test procedure, all of the DUT's functions and features are included in the document, so the entire system can be checked thoroughly.

Therefore, it is very important that the state machine covers all possible states and transitions of the DUT. The NMA switch is not a complex electromechanical device; therefore, the testing procedure's steps are minimal. The NMA switch is now ready to be validated using the following testing procedures.

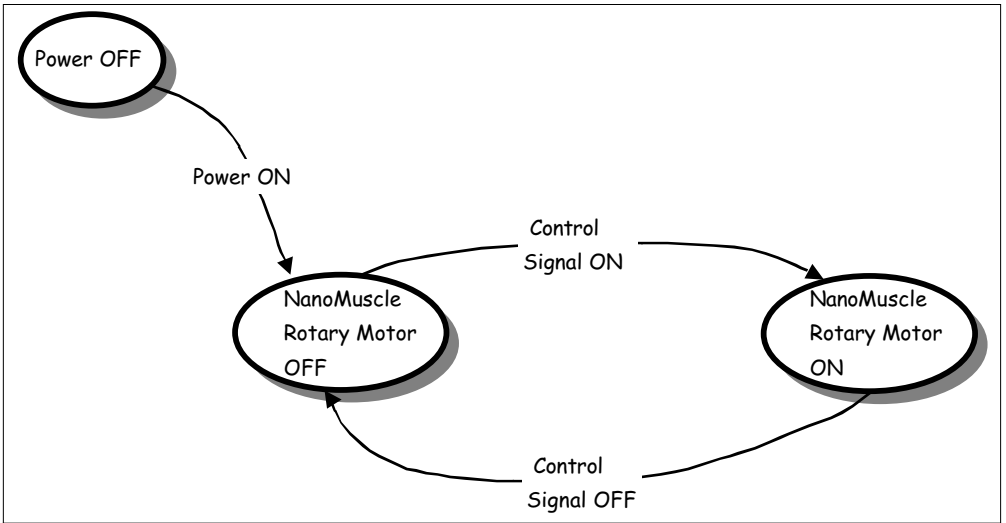


Figure 6. NMA Switch state machine.

Test Procedure Objective:

The objective of this test procedure is to validate the function of the electronic controller circuit and the mechanical assembly of the NMA switch.

1. No power should be applied to the electronic controller circuit for the NMA switch.

2. Apply +5 VDC power supply to the electronic controller circuit.

3. The NMA rotary motor should be off.

4. Press and hold the switch (S1) of the electronic controller circuit.

5. The NMA rotary motor should now be on.

6. Release the switch (S1) of the electronic controller circuit.

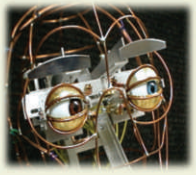
7. Now, the NMA rotary motor should be off.

As the NMA rotary actuator is rotating, observe the mechanical action of the linkage and the movement of the paper clip. This moving electrical contact should touch the fixed binder clip as the attachment head swings down.

If there is any mechanical binding with the NMA switch assembly, make the necessary adjustments to the moving parts to reduce the binding.

With the mechanical assembly of the NMA switch working properly, the

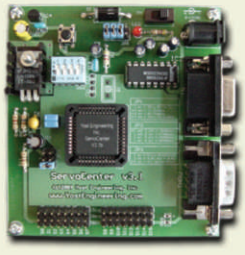
Yost Engineering Inc.
www.YostEngineering.com
1-888-395-9029



Want { More Better Real } Control?

Then try the YEI ServoCenter 3.1 Controller!

ServoCenter allows **independent & simultaneous** serial control of the **seek speed & position** of 16 hobby servos per board.



$48.95 Module Only

$69.95 Full Package

Features:
- 16 Servos per board & up to 16 linked boards for simultaneous control of up to 256 servos!
- Scaled positioning mode for programmable maximum/minimum/startup points.
- Selectable baud rate with a compact yet robust serial protocol.
- On-board regulator provides 6.0V or 4.8V servo power with over-current & thermal protection.
- Code examples for: GCC/LINUX, QBASIC, VB6, VC++6, Turbo C, C#.NET, VB.NET, VC.NET.
- DLL and OCX custom control for instant WIN32 programming.

Package Includes: Power adapter, serial cable, sample code/applications CD, and user's manual / programming guide.

Learn more at: www.YostEngineering.com/ServoCenter

Circle #84 on the Reader Service Card.

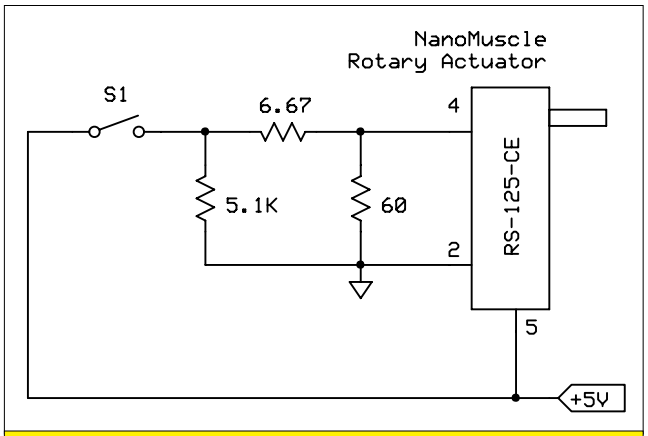


Figure 7. Electronic controller schematic.

homebrew electrical contacts can be tested. Set the DMM to read resistance and attach it to the electrical contacts. If the DMM has an audible feature for doing continuity checks, set the measuring equipment accordingly. Attach the audible ohmmeter to the homebrew electrical contacts and turn on the electronic controller circuit. Press and hold the electromechanical switch (S1) and listen for the audible tone generating from the ohmmeter.

If there is an audible tone, release the switch to silence the audible indicator. Repeat this step several times to assure that the mechanics of the NMA switch and the electronics are working consistently.

If the switch is not producing a tone, turn off the power supply and troubleshoot the device's mechanical and electrical subsystems. Once the problem has been identified and corrected, turn on the +5 VDC power supply and repeat the test procedure.

About the Author

Don Wilcher is an Adjunct Professor at Lawrence Technological University, a technical writer, and an electrical engineer with 18 years of automotive electronics experience. He has written two books, *LEGO Mindstorms Interfacing* and *LEGO Mindstorms Mechatronics*, both published by McGraw-Hill. His papers and engineering education activities can be found on his website: **www.family-science.net/**

Controlling a Small, Electromechanical DC Motor

Controlling a small, electromechanical motor with the NMA switch is quite easy to do. Just as with an electromechanical relay, the external electrical load is switched with its electrical contacts sourced with a series DC power supply.

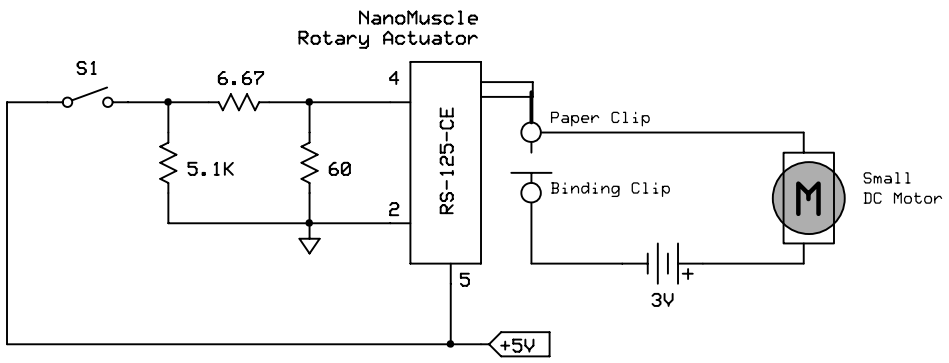
As the NMA switch closes its electrical contacts, the external electrical load powered by the DC power supply will be switched on. The schematic circuit in Figure 8 illustrates how to wire a small electromechanical DC motor to the NMA switch.

A precaution regarding the use of the NMA switch to control external electric loads is that the current requirement should be kept to a minimum of 500 mA. Higher currents might damage the external electrical load or the LEGO pieces that support the homebrew contacts. After wiring the electrical motor to the NMA switch, the entire system can be validated using the test procedure.

The NMA switch — like other electromechanical switching devices — can be used with digital and microcontroller based applications for creating smart products. By electrically interfacing the NMA switch to a computational device and adding a software layer, the MEMs-based component can be programmed for numerous timing and intelligent switching applications, such as smart light switches, robotic object detection, and programmable limit switches.

Hopefully, the project discussed in these two articles has stimulated your creative thinking. The use of this device is only limited by your imagination. **SV**

Figure 8. Schematic for controlling a small DC motor with the NMA Switch.



Robytes

by Dave Calkins

Another month, another collection of robot trivia to amuse your coworkers and annoy your pub-mates. Surely there are more fun stories out there. Got a good story on robots? Email me: news@robotics-society.org If you'd like to get even more robot news delivered to your in-box (no spam, just robo-news) drop a line: subscribe@robotics-society.org

— David Calkins

Look Boss, the Bot! The Bot!

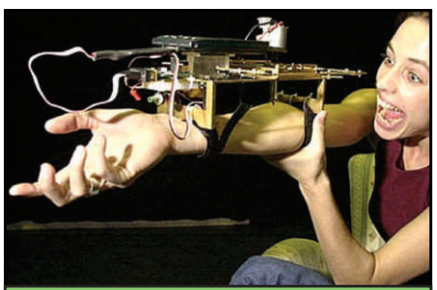


Photo courtesy of Niki Passath.

As you know, I have an affinity for art-bots, but, usually, those are art projects that look like robots. Well, here's a robot that makes its own art! Austrian Niki Passath — a 25-year-old electrician — has invented a robot that will give you a tattoo. Why wait in long lines in seedy parlors to get a tattoo, when you can get one in the privacy of your own seedy home?

Mr. Passath said, "It was a hard job because the only person I could test it on was myself, which was painful, but a good incentive to get it right as soon as possible. [Freddy the robot] is an artist, of course, so he always decides what design the person is going to get, they can't choose, but I haven't had any complaints yet."

Since he's testing his own robot (calling doctors Griffen and Jekyll ...), he has been left with some permanent reminders of the project in the form of some erratic early tattooing attempts. Freddy's computerized brain has a complicated program for creating the designs that are constantly being improved and redesigned before being etched forever and for free on the bodies of volunteers at the trade fair.

Can it cover up my prison tattoo, though?

Roombas Get Modified

Photo courtesy of mondrary.com

Okay, so you want Rosie the Robot, but aren't willing to wait 20 years. You're also not patient enough to teach your Robosapien how to get the vacuum cleaner out of the closet and suck out the dust bunnies from under the couch. What's a roboticist to do?

Well, Ed Paradis and Michael Giambalvo decided to hack their Roomba and turn it into a wireless rover. "We decided the basic requirements were a sturdy, reliable bump switch, wheel encoders, and long battery life. After trying to build something without access to a machine shop or proper tools, we decided just to hack a piece of consumer electronics."

To control the Roomba, they used a PIC16F877 on the Dontronics PicLoader board. The PIC receives serial commands to control the Roomba, report sensor statuses, and provide some level of autonomous control. They've gone from zero to hero pretty quickly, hacking all of the features from the popular autonomous vacuum. For higher level control, they attached a Virgin Webplayer — with two serial ports, a 200 MHz Geode processor, 64 M RAM, and a miniPCI port. They also threw on a 802.11b card, a webcam, and a USB-serial adapter.

Robotic Wheels Keep Rolling Along

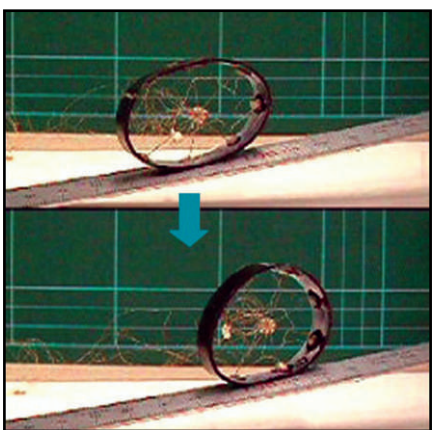


Photo courtesy of Ritsumeikan Univ.

So your robot is stuck on a big old crater on Mars. The nice, round wheels have gotten themselves good and wedged between two Martian rocks. Is the robot doomed? It isn't if it has the shape memory alloy wheels being developed by Ritsumeikan University in Tokyo! Wheels are great for a lot of things, but, well, they're always the same shape and have been for millions of years.

Not only can the new wheels change shape, but they can move under their own power. Rather than using a motor to spin the wheels, the wheels change shape (say, from a circle to an ovoid) and let gravity do the work as they roll forward (much like an upright egg tipping over). Shinichi Hirai and Yuuta Sugiyama, the leads on the project, have designed wheels that can

not only do that, but can also roll uphill by themselves (up to 20 degrees) and even jump on their own — up to twice the height of the wheel itself!

The applications are infinite, but — of course — the military wants them first for scout robots. They could easily be adapted as planetary explorers (saving a lot of weight on the craft, which would result in cheaper space exploration!), but personally, I think it would make a great go-cart.

As if Real Roaches Weren't Bad Enough ...

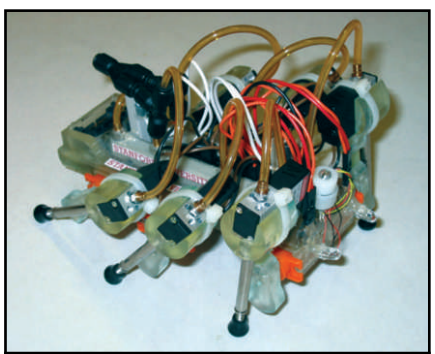


Photo courtesy of Stanford Center for Design Research.

They've been around 280 million years. They could easily survive a nuclear war that would turn you into a chalk stain. They'll eat anything and can live for a month with no head. No, I'm not talking about magazine editors; I'm talking about roaches. (*Speak for yourself, monkey-boy! — Editor Dan*)

Anything so tough has to be worth studying and copying and Stanford has gone ahead and done that. Mark Cutkosky (Director of the Center for Design Research at Stanford) and Bob Full (Biomechanics Professor and Director of the Poly-PEDAL Lab at the University of California at Berkeley) have designed robotic cockroaches that they call "the Sprawl."

The motion of these robots was inspired by cockroach movement. Roaches can run at speeds of up to 50 times their body lengths per second, which is faster than I could even run when being chased by killer droids. The robot versions don't run that fast, but — using cascaded pistons for their motion — they can still do about 15 body lengths/second — which is like you being able to run as fast as a car moving down the freeway.

Unlike Asimo and other bipedal robots that are becoming popular, Sprawl robots have six legs, which make them faster and more durable than our two-legged mechanical clones. The creators keep their sense of humor though and give the robots names like "Sprawley Davidson" and "The Sprawlettes."

I just hope I don't find any in my pantry, eating my crumpets.

Hello KittyBot!

Photo courtesy of Business Design Laboratory.

No, it's not April Fools. I wish that it was. Hello Kitty is once again invading the homes of innocent young Americans and she's bent on brain-washing them into buying more cute, pink things.

Business Design Laboratory — a Tokyo-based robot developer working with Sanrio — has built a Hello Kitty robot to commemorate the 30th anniversary of the ever-popular and

Circle #32 on the Reader Service Card.

oh-so-cute kitty.

Available on November 1 (HK's birthday), the one and a half foot tall robot will retail for about $3,500.00 and weigh under 15 pounds. It can recognize 10 different human faces with its built in camera. Unlike iCybies, Robosapiens, and Aibos, it cannot walk around. It "communicates" with people by using expressions out of 20,000 memorized patterns of conversation while expressing its "emotions" by moving its head and arms. She also has ultrasonic sensors and a speech synthesis program to talk to the little girls whose parents buy it.

The initial run will be 2,000 robots — much like Aibos first run of 2,000. Also like Aibos, orders will be taken through the Internet.

Guess what every little girl under 14 will be asking Santa for?

Robots Get a Leg Over

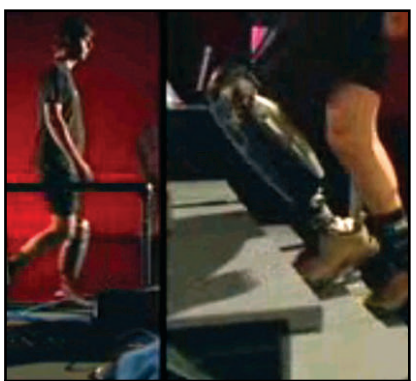


Photo courtesy of Victhom.

Ah, *The Six Million Dollar Man* — I'm sure most of us watched it as kids, and wished that we could have bionic legs that could make us run that fast.

The problem, of course, was that we didn't have the technology to rebuild Steve Austin — not in the 70's — but now we do. Össur (an Icelandic prosthetics company) and Victhom Human Bionics (a German company) recently unveiled the first working bionic leg at the Leipzig REHA — the world's largest international congress for prosthetics, orthotics, and rehabilitation technologies.

Stephane Bedard, the mastermind behind this technology, first envisioned the necessity and potential of highly integrated biomechatronic mechanisms about 10 years ago.

Conventional prostheses are passive, but these bionic devices offer active, automated mobility. They help invigorate and energize amputees because, instead of working against them as they move, they work with them by creating a bi-directional interaction between their bodies and their prostheses.

"We still have a ways to go before we get a bionic man or woman, but this new, powered leg system is an exciting development, offering significant quality of life improvements for the initial group of amputees who will benefit from it," said Hilmar B. Janusson, Vice-President R&D at Össur.

"The founding principle of bionics is the understanding of biological design, learning to imitate natural models that correspond best to the design in process.

Human bionics integrates mechatronic apparatus within the human body, using electronics, computer science, and mechanical technologies to compensate for various physiological or anatomical dysfunctions," explained Stéphane Bédard, Victhom's CSO.

Now, all I can hope for is a date with Jaime Sommers. **SV**

SUPERCIRCUITS www.supercircuits.com

VIDEO CRIME STOPPERS AND JAW DROPPERS!!!

Supercircuits delivers everything you need to catch lowlife criminals red handed, including wireless video, covert video, internet video, and more. From the most economical $12 micro camera to advanced 16 camera systems, Supercircuits delivers cost effective crime stopping performance and rock solid reliability.

WORLD'S BEST VALUE COVERT PINHOLE VIDEO CAMERA
$11.95
Put it on your R/C plane with our video xmitter for a bird's eye view!

SUPER WIDE ANGLE COLOR INLINE CAMERA
Wide angle lens covers 170° field of view with new low distortion 2.2MM lens
$149.95

TINY MICRO VIDEO TRANSMITTER
Capable of 1000 foot transmission distances and is smaller than your thumbnail.
$189.95

FCC APPROVED WIRELESS CAMERA TRANSMITS UP TO 300 FEET!
$149.95

WORLD'S SMALLEST VIDEO CAMERA
Featured in the Guinness Book of Records!
Covert pinhole lens
$99.95

UNDERWATER VIDEO SYSTEM WITH MONITOR & BATTERY
65 foot cable!
$149.95

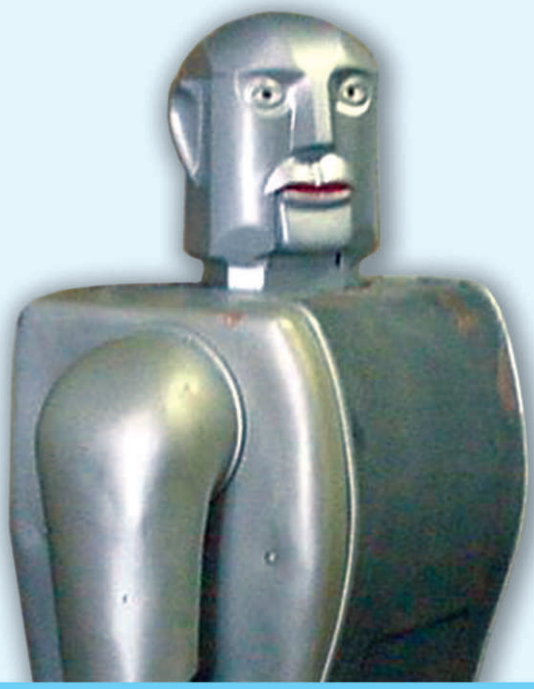
MICRO DIGITAL VIDEO RECORDER
$499.95
Pocket size, records up to 40 hours!

1280 HOUR TIME LAPSE VCR
$179.95

110% LOW PRICE GUARANTEE
*Some restrictions apply. See catalog or website for details.

QUALITY SERVICE INNOVATION 15 Years

1-800-335-9777
1 Supercircuits Plaza
Liberty Hill, TX 78642



The Return of ELEKTRO

by Tom Carroll

There's nothing like an article in a magazine to draw responses from readers. This happened with my article, "Early Robots" in the April 2004 issue of *SERVO*. Many readers asked Editor Dan, "Just where is Elektro now?" Notice the spelling — Elektro, rather than Electro. Most articles and references spell his name Electro, but I received a response from his present owner who calls him Elektro, just as he was called back in his "early years."

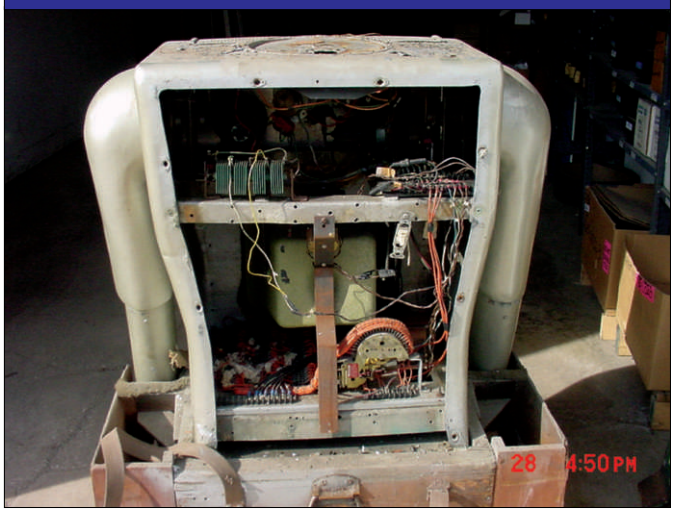
Jack Weeks, who lives in Mansfield, OH, is Elektro's present owner and actually grew up with him. Jack contacted *SERVO* after reading my article. He wanted to give us an overview of the robot's long history. Over several phone interviews and Emails from Jack, he has given me a bit of Elektro's history.

"The Man," as Jack fondly calls him, was created in the Mansfield Westinghouse plant back in 1937-1938. He went on a world tour after the New York World's Fair. Jack's father was one of the designers and builders of Elektro and his dog, Sparko. The robots spent the years around WWII in Jack's father's basement, where Jack played with Elektro.

In the late 40s, he worked with a group rebuilding the robot. Elektro returned to the road with Frank Ruth in 1950 to do promotional tours. During his college years, Jack traveled with The Man, working as Elektro's maintenance man. In 1957, Whit Smith and Jack shipped him to Palisades Park in California as a static display.

Elektro's life then became a bit

Elektro's exposed back, showing years of neglect. Note the round platform with motors for head turning and nodding.



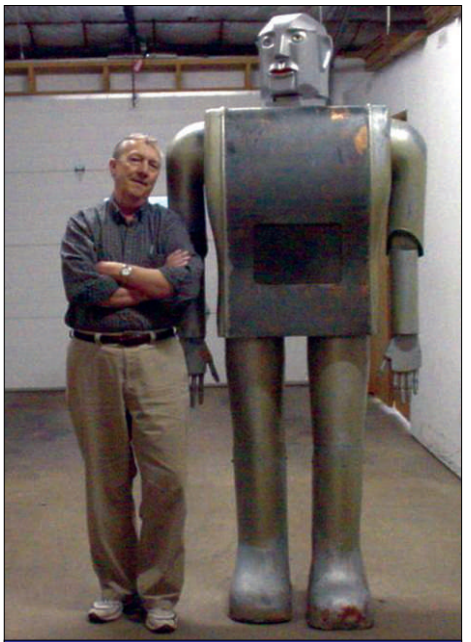
Always a fellow with an eye for a pretty face, Elektro mugs it up with Vicki Matranga. (Vicki is an historian with the International Housewares Association, and a graduate student instructor with Loyola University Chicago.)

interesting, as he was later shipped back to Mansfield, OH, and managed to get "lost." Jack was given his head in 1973 and some drawings in 1990. It wasn't until earlier this year that Jack managed to find the rest of Elektro. He was uncrated on April 23, 2004, and Jack is planning on a full restoration, as the many years have not been kind.

A previous owner painted his aluminum skin silver and glued on a cloth vest. The internal wiring has all hardened and is cracking, but the motors seem fine. Elektro's head is made of cast aluminum. The skin is aluminum and his internal structure is steel.

Jack hasn't found Sparko yet, but believes he is also in Mansfield somewhere. Elektro's head is on display at the Decorative Arts Center in Lancaster, OH, until August 22, 2004, when it is to be returned to the Mansfield Memorial Museum for several months.

His arms and torso — stripped of the cloth vest and adhesive — reveal the original gold paint. He is supported by the original transport caddy.



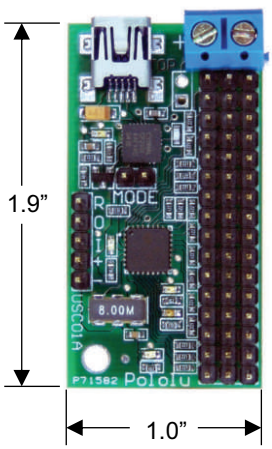
Elektro — out of his storage crates and assembled — poses with Weeks for his first photograph since the 1960s.

Jack has a full set of drawings, copies of most of the literature about Elektro through the years, pins, post cards, sales handouts, and other articles written about The Man. Jack occasionally finds himself "talking" to Elektro, as he says that a distinct personality develops when you're around the robot long enough.

Jack told me that, "It is a creepy feeling, but you can't help yourself from feeling that he is alive." Jack wants his mechanical friend to end up in a good museum somewhere, so that all people can enjoy Elektro as he has for many years. SV

"It is a creepy feeling, but you can't help yourself from feeling that he is alive."

Introducing the Pololu USB 16-Servo Controller



The features you need in a size that fits your robot:

- Control 16 R/C hobby servos
- Dual USB and UART interface
- Independent speed and range for each servo
- 0.5-microsecond resolution
- 50 Hz update rate
- Upgradable firmware
- Small size of 1.9" x 1.0"

Our new servo controller plugs into a USB port and looks like a serial device to your software; when you are ready to disconnect your robot from the computer, the servo controller can also communicate serially with your robot controller.

See www.pololu.com for a complete line of servo controllers starting at only $27.

Pololu Robotics & Electronics

Circle #149 on the Reader Service Card.

PC-BASED OSCILLOSCOPES

PRODUCT	Number of Analog Channels	Number of Input Channels	Maximum Single Channel Sampling Rate	External Trigger	Probe Type	ADC Resolution (bits)	Input Voltage Range (1x Probe)	Voltage Base Range	Time Base Range	Bandwidth
Parallax USB Oscilloscope Parallax, Inc. www.parallax.com	2	2	500 KS/s	Yes	Custom Clips	8	±20 V	100 mV/div to 5 V/div	50 µs/div to 1 s/div	200 kHz
ADC40 Pico Technology Ltd. www.picotech.com	1	0	10 KS/s	No	Any Standard BNC Oscilloscope Style	8	±5 V	±5V	500 µs/div to 50 s/div	10 kHz
ADC-200/20 Pico Technology Ltd. www.picotech.com	2	0	10 MS/s	Yes	Any Standard BNC Oscilloscope Style	8	±20 V	50 mV/div to 20 V/div	100 ns/div to 50 s/div	10 MHz
ADC-212/50 Pico Technology Ltd. www.picotech.com	2	0	25 MS/s	Yes	Any Standard BNC Oscilloscope Style	12	±20 V	50 mV/div to 20 V/div	100 ns/div to 50 s/div	25 MHz
Picoscope 3204 Pico Technology Ltd. www.picotech.com	2	0	50 MS/s	Yes	Any Standard BNC Oscilloscope Style	8	±20 V	10 mV/div to 2 V/div	5 ns/div to 50 s/div	50 MHz
Picoscope 3205 Pico Technology Ltd. www.picotech.com	2	0	100 MS/s	Yes	Any Standard BNC Oscilloscope Style	8	±20 V	10 mV/div to 2 V/div	2 ns/div to 50 s/div	100 MHz
Picoscope 3206 Pico Technology Ltd. www.picotech.com	2	0	200 MS/s	Yes	Any Standard BNC Oscilloscope Style	8	±20 V	10 mV/div to 2 V/div	1ns/div to 50 s/div	200 MHz
SDS 200A softDSP Co., Ltd. www.softdsp.com	2	0	100 MS/s	No	1x/10x BNC 200 MHz Bandwidth Passive Probe	9	42	10 mV/div to 10 V/div	2 ns/div to 10 s/div	200 MHz
DSO-2102M Link Instruments, Inc. www.linkins.com	2	0	100 MS/s	Yes	1x/10x BNC	8	±50 V	20 mV/div to 2 V/div	10 ns/div to 10 s/div	60 MHz
DSO-28464 Link Instruments, Inc. www.linkins.com	4	16	100 MS/s	Yes	1x/10x BNC	8	±20 V	10 mV/div to 2 V/div	5 ns/div to 2,000 s/div	100 MHz
BS310U Bitscope Designs www.bitscope.com	4	8	40 MS/s	Yes	100 MHz x1/x10 switchable oscilloscope probe and Mini Clip Logic Probes	8	±10.4	100 µV/div to 50 V/div	100 ns/div to 10 s/div	100 MHz
BS50U Bitscope Designs www.bitscope.com	3	8	40 MS/s	Yes	101 MHz x1/x10 switchable oscilloscope probe and Mini Clip Logic Probes	8	±10.4	1 mV/div to 50 V/div	100 ns/div to 2 s/div	100 MHz

MATRIX

by Pete Miles

Look for a new comparison matrix every month! Upcoming topics include batteries, drive motors, radio data links and SBCs. Pete is a busy guy, so — if you're a manufacturer of one of these items — please contact him in advance with your product information: BrainMatrix@servomagazine.com

Trigger Modes	Pre/Post Trigger	Coupling	Waveform Generator	Buffer Size (bytes)	Spectrum Range	Continuous Data Logging	PC Interface	External Power Supply	Size (inches)	List Price
Rising, Falling, External	±10%, 50%, and 90%	None	N/A	1,500	0 to 25 kHz	Yes	USB 1.1	None	5 x 2.3 x 1.5	$189.00
10 kHz	-100% to +100%	None	N/A	N/A	0 to 10 kHz	Yes	Parallel Port	None	2.2 x 2.1 x 0.6	$129.00
10 MHz	-100% to +100%	AC and DC	5 to 250 kHz	8K	0 to 10 MHz	Yes	Parallel Port	12 V @ 500 ma	7.3 x 5.3 x 1.3	$557.00
25 MHz	-100% to +100%	AC and DC	5 to 250 kHz	128K	0 to 25 MHz	Yes	Parallel Port	12 V @ 500 ma	7.3 x 5.3 x 1.3	$930.00
Rising, Falling, External	-100% to +100%	AC and DC	1 kHz	256K	1 to 25 MHz	Yes	USB 2.0	None	7.3 x 5.3 x 1.3	$744.00
Rising, Falling, External	-100% to +100%	AC and DC	1 MHz	512K	1 to 50 MHz	Yes	USB 2.0	None	7.3 x 5.3 x 1.3	$1,116.00
Rising, Falling, External	-100% to +100%	AC and DC	1 MHz	1M	0 to 100 MHz	Yes	USB 2.0	None	7.3 x 5.3 x 1.3	$1,489.00
Rising, Falling, Pulse Trigger, Delay Trigger	±4 Divisions	AC, DC, GND	N/A	Up to 512K	0 to 200 MHz	Yes	USB 1.1	None	5.1 x 4.4 x 1.5	$975.00
Rising, Falling, Adjustable, TV-H, TV-V, Pulse Width, Pulse Count	10 Divisions in 0.04 increments	AC, DC, GND	N/A	32K	50 MHz	Yes	USB and Parallel Port	7.5 V @ 1 amp	7 x 3.8 x 1.0	$650.00 parallel port, $725.00 USB
±Slope, Adjustable	10 Divisions in 0.04 increments	AC, DC, GND	N/A	128K	100 MHz	Yes	IBM ISA Slot	None	N/A	$4,000.00
Analog Level/Edge Rising/Falling, Logic State, False/True, True/False, Analog/Logic cross-triggering (all modes), 5 µS filter and/or adjustable filter	Pre-trigger hold-off adjustable, Post Trigger delay (8 µS to 2 hours)	AC, DC, GND	10 MHz Analog	128K	100 MHz	Yes	USB 2.0	12 V @ 500 ma	3.9 x 5.9 x 2.2	$495.00
Analog Level/Edge Rising/Falling, Logic State, False/True, True/False, Analog/Logic cross-triggering (all modes), 5 µS filter and/or adjustable filter	Pre-trigger hold-off adjustable, Post Trigger delay (8 µS to 2 hours)	AC, DC, GND	10 MHz Analog 40 MHz Logic Pattern	32K	100 MHz	N/A	USB 2.0	9 V @ 290 ma or USB	3.9 x 3.0 x 1.0	$295.00

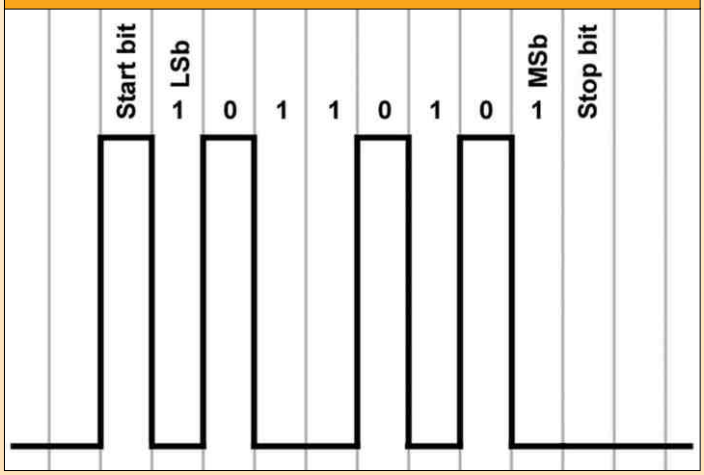
Last month, it was promised that this column would discuss something that was more immediately usable in robotic projects. Any robot that is of moderate sophistication will need to transfer data from one processor or component to another. This month's column will discuss different strategies to send your data from one place to another.

Different formats have different strengths. Sometimes, you will find that you cannot specify which format you will need to use since you will have to use a very specific chip that communicates with only one format. Because of situations like that, it is useful to have an understanding of all the common formats.

Let's start with sending your data in parallel. This is often done when the data absolutely needs to get to where it is going as fast as possible. When you are sending your data in parallel format, you are sending it through multiple wires simultaneously. Each bit sent will have its own wire. Often, you will be using wires in multiples of eight that represent one or more bytes.

One thing that must be considered is that you must signal to the receiving device that you are showing new data. This necessitates an additional signal line to indicate that new data is ready to be read. This line is often called the strobe line. To send your data, you will first display the data on the parallel signal lines and then raise the strobe line high to signal to the receiving device that the data is ready to be read. After that, you will lower the strobe line. Care must be taken to ensure that the receiving device has enough time to detect the high pulse on the strobe line before you drive it low again.

Figure 1. RS232 data — note the inverted signalling.



This month, you will find that many chips are using a serial data format to do their communication. When a chip is communicating in serial format, it sends data bits — one after the other — on the same wire. Serial communication can be divided into three methods. The first is asynchronous communication. Asynchronous data is sent without a clock signal. A common format that is used is the RS232 data format. The RS232 specification specifies voltage levels, but this column will ignore that part of the format.

RS232 allows for five to eight data bits and it requires you to have start and stop bits on either side of your data bits. You will always have one start bit, but can have more than one stop bit. More specifically, the RS232 specification allows you to have one, one and a half, or two stop bits. The RS232 protocol allows for a parity bit to be added to each group of data bits.

Because of all of these options, the RS232 protocol has a lot of flexibility and could conceivably be tricky to implement. The good news is that, for the most part, everyone has ignored all of the options available and uses just one. This form is commonly called 8N1 format. This stands for eight data bits, no parity bit, and one stop bit. Figure 1 shows a byte in RS232 8N1 format. Note that the data bits are inverted. RS232 specifies that the bits be inverted for some reason.

The second form of serial data could be described as hybrid. Hybrid serial data could be described as serial data with an embedded clock signal. One common method of sending data that has a clock encoded into it is to send it with Manchester encoding. Manchester encoding sends two bits for each data bit.

If you want to transmit the value of zero, then you send "01" and, if you want to transmit a one, then you will send "10." The end result of this strategy is that you will never have the serial data remain high or low for more than two clock cycles. The receiving system will be able to recover the clock

and the data from the same signal line.

The final form of moving serial data is called synchronous. Synchronous serial data has a separate clock signal line. This form of serial data is the most commonly used form of serial data that is used for chip-to-chip communication.

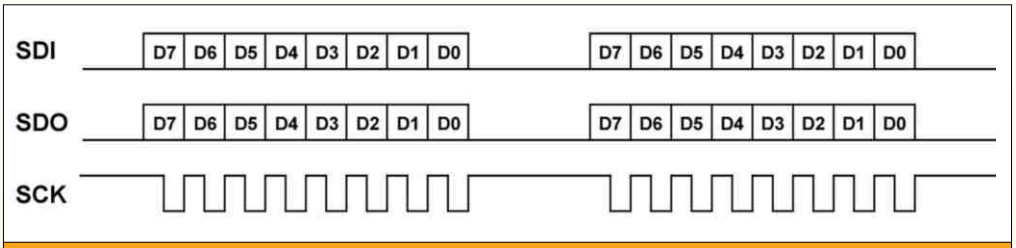


Figure 2. SPI clock and data waveforms.

The first format that we will discuss here is the SPI (Serial Peripheral Interface) format. SPI is a format that uses a clock signal, a serial data in signal, a serial data out signal, and, often, a chip select signal. SPI devices clock their data on either the rising or falling edge of the clock signal.

The device being connected to determines the edge used. SPI devices are also specified to be either a master or slave SPI device. If the device is set up to be a master, then it generates the clock signal. SPI is very easy to use. If you are a fan of the BASIC Stamp, you can talk to SPI devices using the SHIFTIN and SHIFTOUT commands.

Figure 2 shows the waveform for a device that is set up to send and receive data on the falling edge of the clock and that has the clock idle high. As you can see, when sending data through SPI, you can simultaneously send and receive data. This is not necessary, though.

Some processors, such as some of Microchip's PIC processors, have SPI peripherals built into the chip. Take a look at Figure 3 for an example of how to connect a PIC16F873 to the MPC3208, which is an eight channel, 12-bit analog-to-digital converter. Pin 10 of the MPC3208 could be connected to any I/O pin on the PIC. It is the chip select line that tells the MCP3208 whether to respond to commands or not. The other three connections are connected so that the SPI peripheral in the PIC can do all of the communication and leave the processor to do other things.

Just connecting one MCP3208 to this particular PIC is not so useful, since this PIC already has a six channel, 10-bit analog-to-digital converter built in. The big advantage is more apparent if you want a lot of analog to digital converters connected to the PIC. You can add as many as you like. The only thing that you would need to do is to have a separate chip select line running to each MCP3208.

Sometimes, it is necessary to connect multiple chips using SPI to the same serial line, but the chips are capable of driving the line both high and low and they can't be set to be Hi-Z when they are not in use. This situation would cause the chips to fight each other, so you will need to do something to prevent that.

A simple solution to this is shown in Figure 4. By adding a diode to each serial output, you make it so that the output pin can only drive the serial line high. A pull down resistor drives the line low when none of the SPI devices are outputting a high level.

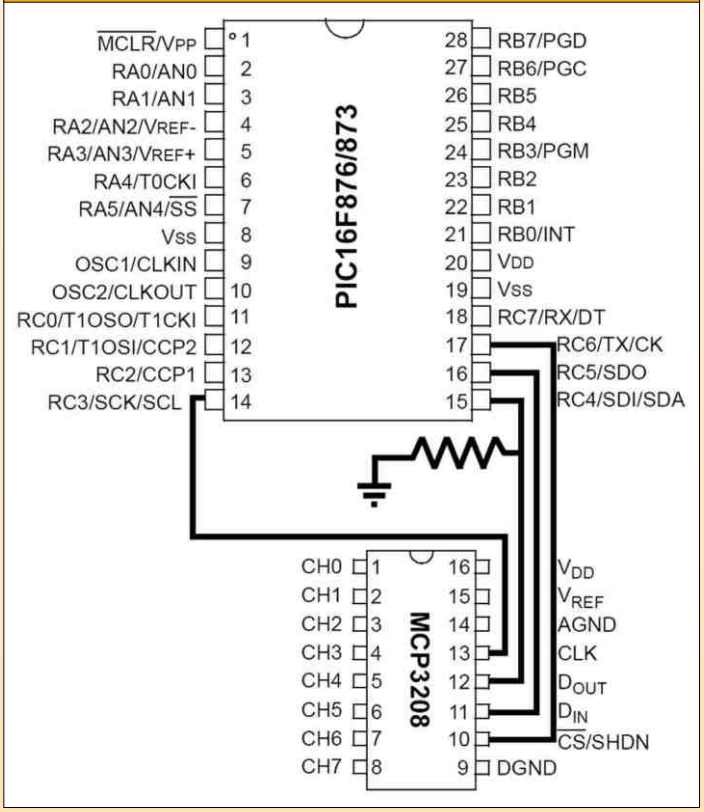
Another commonly used serial format that is in use today is the I^2C format. I^2C stands for Inter-Integrated Circuit. I^2C is more complicated than SPI, but allows for multiple devices to be connected to the I^2C bus. It does not require that each device use a chip select input.

Instead, I^2C devices each have an address that they will respond to. This is good for systems that have many I^2C devices — maybe different types of devices — connected to them. I^2C uses only two signal lines: clock (SCK) and serial (SDA). Both of these lines are bi-directional, which means that any device on the I^2C network can be a master device, a slave device, or both.

To communicate with an I^2C device, a start signal is sent, followed by a seven-bit address and a single bit to indicate if the master is requesting a read or a write. If this bit is high, then it will be a read operation; otherwise, it will be a write operation. After that, an acknowledge signal is sent back by the slave device with the requested address.

At this point, if the operation is a read operation, then the slave device will transmit eight bits of data and the

Figure 3. Connecting a PIC to an MCP3208.



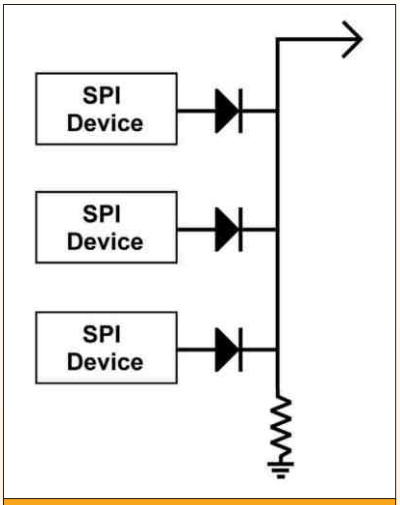


Figure 4. How to connect multiple chips to drive high and low on SPI.

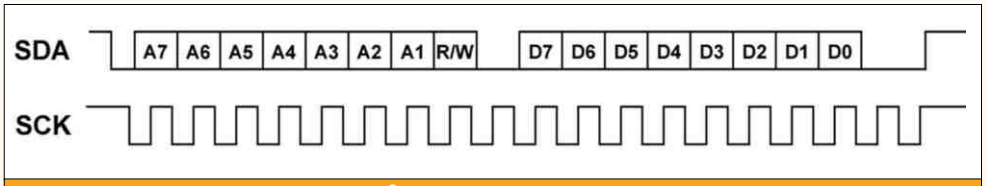


Figure 5. I²C clock and data waveforms.

master will respond with an acknowledge signal. The slave device will continue sending data until the master does not send an acknowledge and, then, the master sends a stop signal. If the operation was a write, the master sends eight bits of data, after which, the slave device will respond with an acknowledge. This will continue until the master has sent all of its data. After the slave device sends its last acknowledge, the master device will send a stop signal.

Let's go over these commands in more detail. To send a start command, a processor or other device will first lower the serial line and, then, lower the clock line. To send a stop command, the clock line will be raised, followed by the serial line. For the receiving device to send an acknowledge (ACK) signal, it will wait until the master sends a low clock. At that time, it will pull the serial line low. The master will raise the clock high and then send it low again. At this point, the receiving device will release the serial line and let it go high once again.

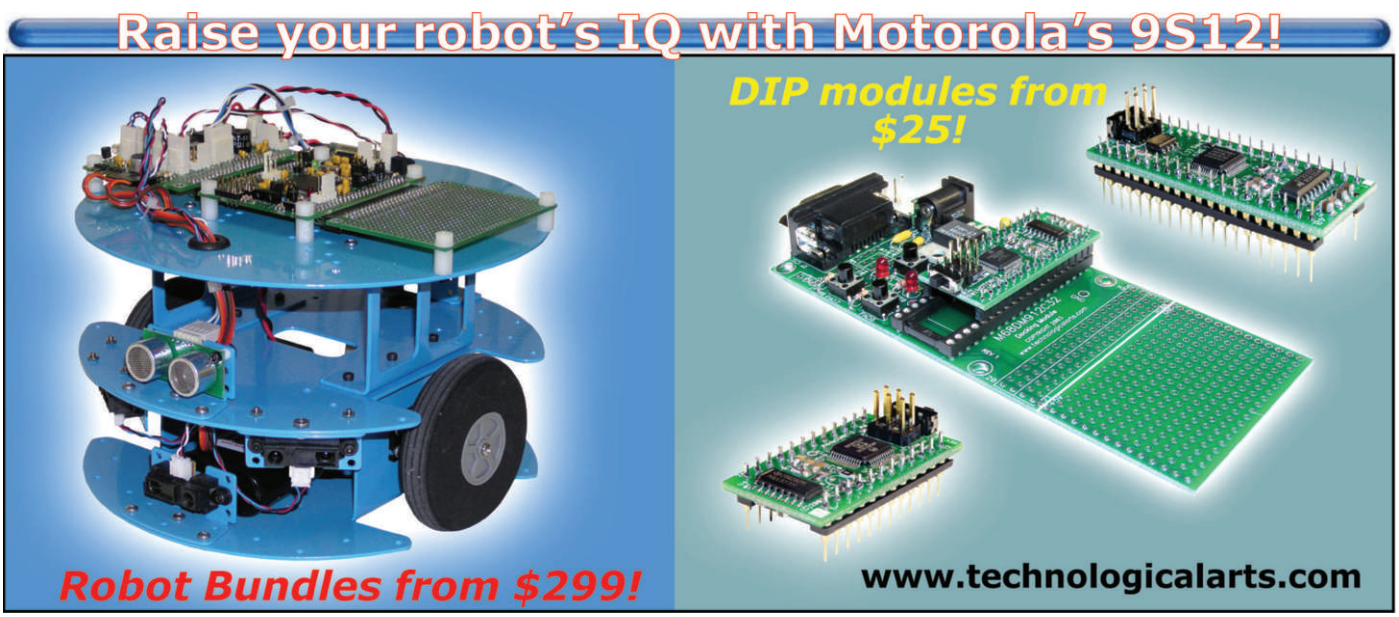
When a slave device is transmitting data to the master, it writes its data on the falling edge of the clock and the master reads the data on the rising edge. This is the same as when the master transmits data to a slave device. In that case, the master writes its data to the serial line just after it drives the clock line low. The slave device will read the data on the rising edge of the clock. In both cases, the most significant bit of the byte that is being sent is transmitted first.

Let's take a look at what these waveforms look like. In Figure 5, the first thing that happens is that the master device will pull the serial line low, followed by the clock line. This is the start signal. After that, it sends the seven address bits, followed by the read/write bit. In this case, we will assume that the read/write bit was a 0, which signifies a write operation.

The master device releases control of the serial line and the slave responds with a low for one clock cycle, which represents the acknowledge signal. After that, the master sends a byte of data and the slave responds with another acknowledge. Finally, the master sends the stop signal, which is where the clock goes high and, then, the serial line goes high.

Author Bio

When not writing for *SERVO Magazine*, Jack runs Buffington Effects, a company that designs and builds animatronics and motion control devices for the entertainment industry. Check out his website at **www.BuffingtonFX.com**



When using I²C, you can send as many bytes of data as you like. The example in Figure 5 just happened to send only one. If the operation was a read operation, then — after the initial address and read/write bit — the master would be sending the acknowledge signals and the slave would be sending the serial data instead of the master.

Take a look at Figure 6 to see how to connect a 24LC515 — which is a 64 kilobyte serial EEPROM that uses I²C — to a PIC16F873 microcontroller.

There is one final method of communicating between different devices. This method is sometimes referred to as pulse width modulation (PWM). Pulse width modulation for data transmission is generally used in two situations. The first is when accuracy is not critical. This might be the case where you need a general idea of how large a value is. The second case is where the receiving device, the sending device, or both are analog circuits that can't interpret serial or parallel data. Analog devices can use pulses of different lengths to come up with an analog voltage level, which can then be used by the circuit.

Pulse width modulation is pretty much what it sounds like. The transmitting device will raise the signal line for an amount of time — which varies — and then lower it again. The length of time that the signal line is high determines the value that is being sent.

For example, two processors may communicate using PWM where a pulse that is 0.1 milliseconds long represents a value of 0 and a pulse that is 2.0 milliseconds long represents a value of 1000. It would be the responsibility of the receiving processor to determine how long the pulse was and to convert that back into a number value between 0 and 1000.

Sending data between different devices is an important part of working with robotics. It is often easier to have processors near the devices and sensors that they are controlling than to have everything located on one circuit board with dozens of wires coming out of it leading to the various devices.

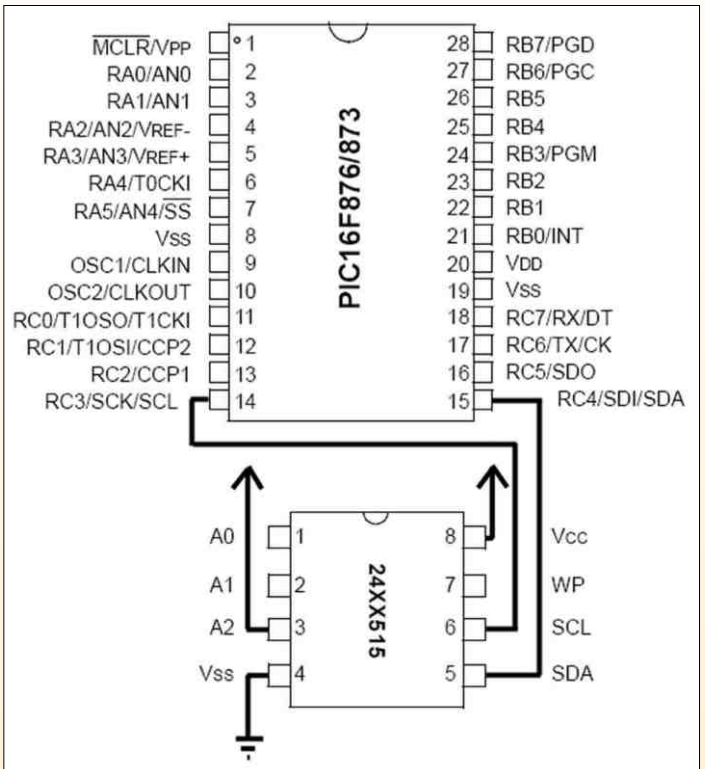


Figure 6. Connecting to a serial EEPROM using I²C.

This article described how to communicate with devices using a variety of different methods of communication. Using what you read here should be enough to get you started with these formats, but you will always need to read the data sheet of the part that you are trying to communicate with to find out the specifics of how to send it the proper data it needs in order to operate. **SV**

Circle #118 on the Reader Service Card.

CORDIC Algorithms

Trigonometric Functions for Limited Hardware

by Bill Brew

So, what do you do if you need to compute the value of a trig function and all you have is a microcontroller with none of those hand-dandy math libraries that are available on "big" computers? You make your own! You can probably think of a few ways to do this, for example, with table lookup or Taylor series polynomials. Another possibility that you might not know about is a CORDIC algorithm.

CORDIC (from **CO**ordinate **R**otation **DI**gital **C**omputer) algorithms are a family of computational procedures oriented around vector rotations. A frequent use of CORDIC algorithms is to compute trigonometric functions, although they have many other uses.

CORDIC algorithms were first described by Jack Volder in 1959 and have been used in many diverse situations over the years. More recently, CORDIC algorithms have been considered for implementation of trigonometric functions in FPGAs.

Many robot geometries involve the use of vector rotations and trigonometric functions in their solutions. For example, I have used a CORDIC algorithm in determining the wheel drive speeds for a three wheel holonomic robot similar to the Palm Pilot Robot. Determining the joint angles necessary to position a robotic arm involves trigonometry and vector operations.

In this article, I will derive the equations underlying CORDIC algorithms and illustrate their use with an example. I will also discuss practical considerations for implementing CORDIC algorithms and include source code for several implementations. CORDIC algorithms are very straightforward and basic high school trigonometry is all that is necessary to understand them.

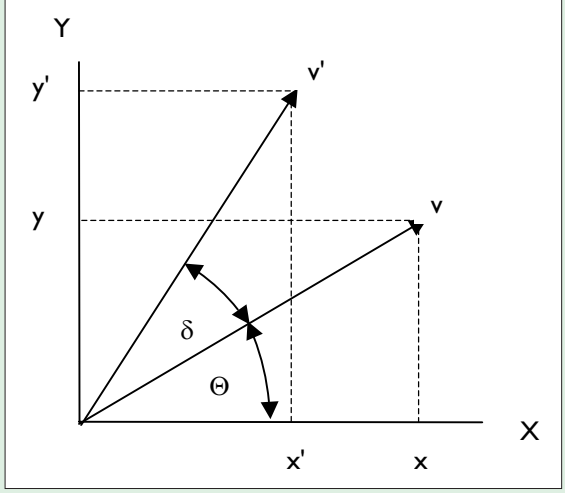
The Basic Idea

The inner loop of the CORDIC algorithms is a series of vector rotations using carefully chosen incremental angles. Each rotation angle is picked so that the vector components after the incremental rotation can be computed using only a shift and add. The rotation angles are of decreasing magnitude and, by choosing to rotate by either ± the incremental angle, a total rotation of arbitrary resolution can be achieved.

Based on the core ability to rotate a vector, variations of the CORDIC algorithms can compute trigonometric functions and coordinate transformations.

The fact that CORDIC algorithms essentially use only shift and add makes them well suited to situations where there is limited support for multiplication, for example, with microcontrollers that have limited hardware multiply.

Figure 1. Vector rotation by angle δ.



Mathematical Foundation

The starting point for understanding CORDIC algorithms is to understand how the components in a vector rotation are computed. Figure 1 illustrates a vector rotation by an angle δ.

I'll start with the following equations for the components of the rotated vector, x′ and y′, in terms of the original components

of the vector, x and y. If you are interested in learning how they are derived, see the sidebar "Vector Rotation 101."

$$x' = cos(\delta) \bullet \left[x - y \bullet tan(\delta) \right]$$

$$y' = cos(\delta) \bullet \left[y + x \bullet tan(\delta) \right]$$

CORDIC algorithms involve iteratively rotating a vector to reach a desired total rotation angle. Therefore, these equations can be viewed as telling us how to compute the x and y components of the vector on iteration *i+1* in terms of the values on iteration *i*.

Each iteration will rotate by either plus or minus δ_i. In the basic algorithm, the choice of a ± rotation is determined by comparing the current cumulative rotation angle to the desired total rotation angle. If the current angle is less than the desired angle, then rotate positive; otherwise, rotate negative. I will use the symbol Δ_i to represent the cumulative rotation angle.

The equations can now be re-written in the following form:

$$x_{i+1} = cos(\pm\delta_i) \bullet \left[x_i - y_i \bullet tan(\pm\delta_i) \right]$$

$$y_{i+1} = cos(\pm\delta_i) \bullet \left[y_i + x_i \bullet tan(\pm\delta_i) \right]$$

$$\Delta_{i+1} = \Delta_i \pm \delta_i$$

The key idea underlying the CORDIC approach to vector rotation is to realize that if the rotation is going to be done in increments, then the incremental angles can be picked to make the computation simpler. In particular, if the angles are picked so that the value of the $\tan(\delta_i)$ term is 2^{-i}, then the multiplication of x and y by $\tan(\delta_i)$ can be approximated using a simple right shift. Table 1 illustrates the incremental rotation angles that satisfy this property.

One more observation will get us to the final rotation equations; we need to do something with the $\cos(\delta_i)$ terms. By employing the trigonometric identity $\cos(\alpha) = \cos(-\alpha)$, the multiples by $\cos(\delta_i)$ on each iteration can be delayed, since the effect is independent of whether the rotation was positive or negative. Instead, the $\cos(\delta_i)$ multiples can be saved up and (conceptually) there is one combined multiple at the end by a constant that is dependent only on the number of iterations. I will call this constant G_i. As you can see from Table 1, the value of Gi converges rapidly to approximately 0.60725.

Putting these observations together gives us the final equations:

$$x_{i+1} = x_i - y_i \bullet \pm 2^{-i}$$
$$y_{i+1} = y_i + x_i \bullet \pm 2^{-i}$$
$$\Delta_{i+1} = \Delta_i \pm \delta_i$$

We now have an algorithm to rotate a vector where each iteration involves only a shift and add. There might be one final multiplication by G needed, but — in practice — this usually doesn't add any additional overhead. For example, you can completely skip the multiply when computing sine and cosine. In many applications, you are going to scale the result of the rotation by some other constant, anyway, for example, the wheelbase of the robot. In this case, you can pre-adjust the other constant to compensate for G and, again, no additional multiply is needed.

One final point: The algorithms as described above only work for angles between ±90°. To work with arbitrary angles, two additional steps are needed. First, the magnitude of the angle is reduced to less than 180° by repeatedly subtracting 360°. Second, if the resulting angle is greater than ±90°, then a trivial initial rotation by ±90° is performed to bring the angle into range. I'll describe this a bit more below.

Variations on the Algorithms

The equations I derived above can be used in different ways. There are two modes of operation depending on how the direction of rotation is decided. I will call them rotate by angle mode and rotate by component mode. For each mode, there are variations.

Rotate by angle mode is what I have been discussing up to now; look at the angle accumulator and choose the rotation direction so that you converge to the desired angle. Some specific variations include:

• *Compute sine and cosine*: Set the initial x component to G and the initial y component to 0. Rotate by the desired angle. The final x component is the cosine of the angle and the final

i	$\tan(\delta_i)$	$\delta_{i(degrees)}$	$\cos(\delta_i)$	G_i
0	1.00000	45.00000	0.70711	0.70711
1	0.50000	26.56505	0.89443	0.63246
2	0.25000	14.03624	0.97014	0.61357
3	0.12500	1.12502	0.99228	0.60883
4	0.06250	3.57633	0.99805	0.60765
5	0.03125	1.78991	0.99951	0.60735
6	0.01563	0.89517	0.99988	0.60728
7	0.00781	0.44761	0.99997	0.60726
8	0.00391	0.22381	0.99999	0.60725
9	0.00195	0.11191	1.00000	0.60725
10	0.00098	0.05595	1.00000	0.60725
11	0.00049	0.02798	1.00000	0.60725
12	0.00024	0.01399	1.00000	0.60725
13	0.00012	0.00699	1.00000	0.60725
14	0.00006	0.00350	1.00000	0.60725
15	0.00003	0.00175	1.00000	0.60725
16	0.00002	0.00087	1.00000	0.60725
17	0.00001	0.00044	1.00000	0.60725

Table 1. CORDIC angles and gain.

y component is the sine of the angle. Note that this computes sine and cosine simultaneously.

• *Polar to Cartesian conversion*: Set the initial x component to the polar magnitude times G and the initial y component to 0. Rotate by the desired angle. The final x and y components are the Cartesian components of the original polar vector.

In **rotate by component** mode, you rotate by looking at one of the vector components and choosing the rotation direction to make it converge to 0. Some specific variations include:

• *Compute arctangent* (given x and y components, find the corresponding angle): Set the initial x and y components. Rotate to make the y component converge to 0. The final angle accumulator value is the arctangent.

• *Cartesian to polar conversion*: Set the initial x and y components. Rotate to make the y component converge to 0. The final angle accumulator value is the polar angle. Multiply the final x component by G to get the polar magnitude. (This is much simpler than the square root of the sum of the squares approach if you are using a microcontroller.)

These are just a sample of the functions that can be computed with CORDIC-based algorithms. With straightforward modifications, they can be extended to arcsine, arccosine, hyperbolic functions, and many others.

Vector Rotation 101

Figure 1 illustrates a general vector rotation. We need to figure out how to compute x' and y' from x, y, and δ.

From the basic definitions of sine and cosine, we know that:

$$x = |v| \cos(\theta) \qquad x' = |v| \cos(\theta + \delta)$$
$$y = |v| \sin(\theta) \qquad y' = |v| \sin(\theta + \delta)$$

Using the trigonometric identities for sine and cosine of the sum of two angles, the equations for x′ and y′ can be re-written as:

$$x' = |v| \bullet [\cos(\theta) \bullet \cos(\delta) - \sin(\theta) \bullet \sin(\delta)]$$
$$y' = |v| \bullet [\sin(\theta) \bullet \cos(\delta) + \cos(\theta) \bullet \sin(\delta)]$$

Now, simplify using the equations for x and y to get:

$$x' = [x \bullet \cos(\delta) - y \bullet \sin(\delta)]$$
$$y' = [y \bullet \cos(\delta) + x \bullet \sin(\delta)]$$

One last transformation using the identity $\tan(\alpha) = \sin(\alpha)/\cos(\alpha)$ gives us the starting point for the CORDIC equations:

$$x' = \cos(\delta) \bullet [x - y \bullet \tan(\delta)]$$
$$y' = \cos(\delta) \bullet [y + x \bullet \tan(\delta)]$$

An Example

Well, enough of this theory. Let's take a look at how this really works. To illustrate the operation of a CORDIC algorithm, I will step you through a sine, cosine computation. To keep things simple, I will use real numbers with two decimals. (Yes, I know your microcontroller doesn't have real numbers; more on this below.) I will use nine iterations, i = 0 .. 8, to compute the sine and cosine of 120°. Table 2 shows each step in the computation.

For i = 8, the value of G_i is 0.60725. So, the initial x component is 0.61 and the initial y component is 0.

Since the initial angle is greater than 90°, the first step is a special +90° rotation to bring the remaining angle into range. This is easy; just make x′ = y, y′ = x, and $\Delta' = \Delta + 90°$. The remaining rotation is 30°, which is within our range.

The rest of the computation proceeds in a straightforward manner, following the equations described above. At each step, we decide whether do a positive or negative rotation so that the angle accumulator moves towards 120°. With this decided, the equations tell us how many times to shift and whether to add or subtract. After nine iterations, the angle accumulator converges to 120.04°. The final x component, -0.50, is the cosine and the final y component, 0.87, is the sine.

Practical Considerations

Now that you have a basic understanding of the algorithms, let's talk about some practical considerations for implementing CORDIC algorithms on a microcontroller.

As you can see above, the natural expression of the algorithms is in terms of real numbers. In a big computer, these would be represented using floating point numbers. Your microcontroller likely only supports integers. So, you will need to use scaled integers where you implicitly assume a binary point between two of the bits in the number.

Next, let's consider the representation used for angles. In the discussion so far, I have used degrees as the unit for angles. However, the CORDIC algorithms are indifferent to the units for angles, so there is a choice. Degrees are a common unit for angles, but are really rather arbitrary. Radians are a more natural unit and are widely used in mathematics and engineering. Another angle unit that you should consider for low level controllers is fractional revolutions. Fractional revolutions are natural, if, for example, you are working with shaft encoders that give 2^n counts per revolution. In this case, you represent angles using scaled integers, where the binary point is *n* bits from the right. This is the representation I have used in the example code.

You also need to decide the number of iterations to perform. Each iteration narrows the angular error by the corresponding incremental CORDIC angle. You will usually want enough iterations so that the final incremental rotation is about the same as the least significant bit in your angle representation. For example, if your system allows resolving angles to 512 parts per revolution, you need to do at least

iteration number (i)	Δ_i (degrees)	rotation needed (degrees)	δ_i (degrees)	Δ_{i+1} (degrees)	x_i	y_i	shifted x_i	shifted y_i	x_{i+1}	y_{i+1}
Initializers	0.00	120.00	90.00	90.00	0.60	0.00			0.00	0.61
0	90.00	30.00	45.00	135.00	0.00	0.61	0.00	0.61	-0.61	0.61
1	135.00	-15.00	-26.57	108.43	-0.61	0.61	-0.30	0.30	-0.30	0.91
2	108.43	11.57	14.04	122.47	-0.30	0.91	-0.08	0.23	-0.53	0.83
3	122.47	-2.47	-7.13	115.35	-0.53	0.83	-0.07	0.10	-0.43	0.90
4	115.35	4.65	3.58	118.92	-0.43	0.90	-0.03	0.06	-0.48	0.87
5	118.92	1.08	1.79	120.71	-0.48	0.87	-0.02	0.03	-0.51	0.86
6	120.71	-0.71	-0.90	119.82	-0.51	0.86	-0.01	0.01	-0.50	0.87
7	119.82	0.18	0.45	120.26	-0.50	0.87	0.00	0.01	-0.50	0.86
8	120.26	-0.26	-0.22	120.04	-0.50	0.86	0.00	0.00	-0.50	0.87

Table 2. CORDIC Computation of sine and cosine of 120°.

eight iterations; one count = 360/512 = 0.703° — Table 1 tells us this is between seven and eight iterations.

Depending on how you are going to use the CORDIC algorithms, you will likely need to use scaled integer representations for vector components, gain constants, etc. There are many variations here, so I won't attempt to go into the details. Just carefully think through the cases and make sure to allocate enough fraction bits to maintain the resolution that you need. Also, watch out for overflow situations.

Sample Implementations

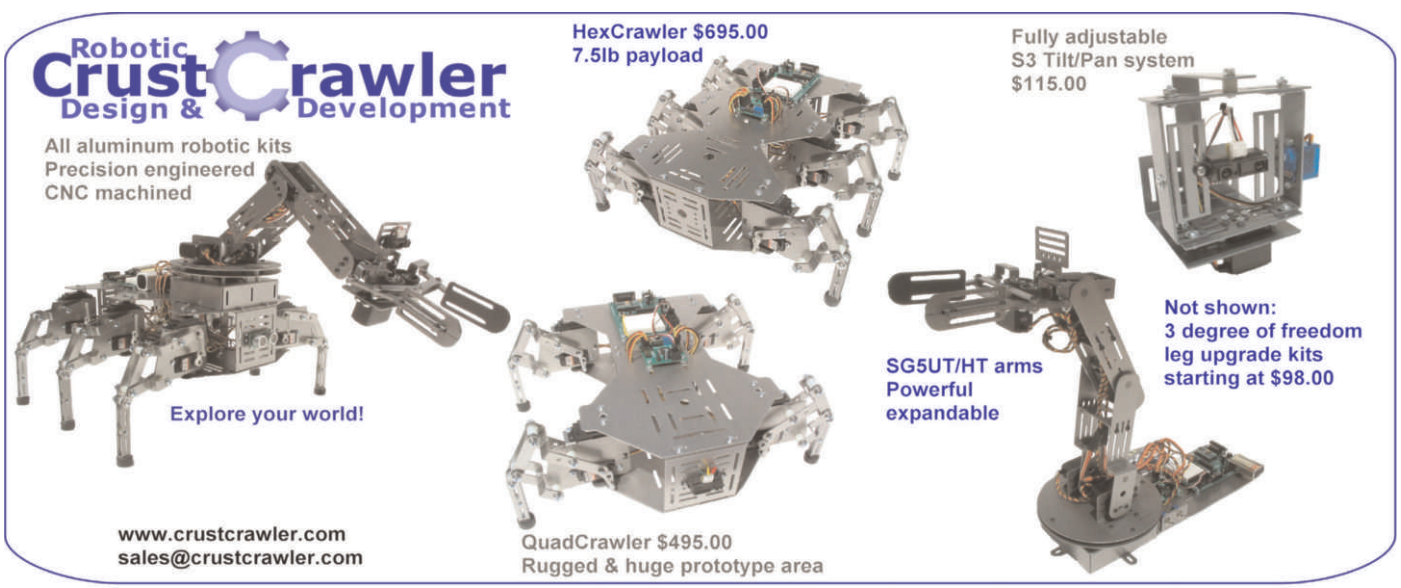
To get you started, I have written sample C implementations of several variations on the CORDIC algorithms:

- Vector rotation
- Sine and cosine
- Polar to Cartesian conversion
- Arctangent
- Cartesian to polar conversion

The code concretely illustrates the algorithms and can serve as a starting point for your implementation. The source code is available from the *SERVO Magazine* website, **www.servomagazine.com**

Conclusion

You should now have a good understanding of CORDIC algorithms. You should at least have a basic understanding of their mathematical foundation. More importantly, you should also understand their applicability, in addition to being well equipped to tailor them to your situation. I have found CORDIC algorithms useful several times and I hope that I have equipped you with enough knowledge and curiosity that you will find them useful, also. SV



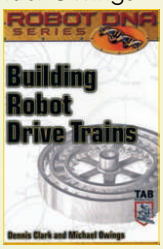
Parallax is the exclusive reseller of our QuadCrawler and HexCrawler Robot Kits (www.parallax.com)

Circle #55 on the Reader Service Card.

Mind Candy For Today's Roboticist

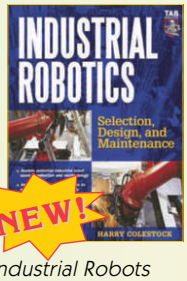
The SERVO Bookstore

Building Robot Drive Trains
by Dennis Clark / Michael Owings



This essential title in McGraw-Hill's *Robot DNA Series* is just what robotics hobbyists need to build an effective drive train using inexpensive, off-the-shelf parts. Leaving heavy-duty "tech speak" behind, the authors focus on the actual concepts and applications necessary to build — and understand — these critical, force-conveying systems. **$24.95**

Industrial Robotics
by Harry Colestock

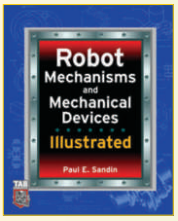


With so many industries taking advantage of the tremendous advances in robotics, entities ranging from small family businesses to large corporations need assistance in the selection, design, set-up, maintenance, and economic considerations of industrial automation. *Industrial Robots* shows how to achieve maximum productivity with robotics, classifies robots according to their complexity and function, and explains how to avoid common automation mistakes. **$39.95**

Robots for Kids
Exploring New Technologies for Learning, First Edition
Edited by Allison Druin / James Hendler

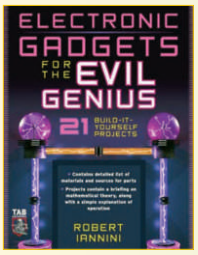
Robots for Kids: Exploring New Technologies for Learning opens with contributions from leading designers and researchers — each one offering a unique perspective into the challenge of developing robots specifically for children. The second part is devoted to the stories of educators who work with children and use these devices, exploring new applications and mapping their impact. Throughout the book, children's essays are provided, discussing their first-hand experiences and ideas about robots. This is an engaging, entertaining, and insightful book for a broad audience — including HCI, AI, and robotics researchers in business and academia, new media and consumer product developers, robotics hobbyists, toy designers, teachers, and education researchers. **$50.95**

Robot Mechanisms and Mechanical Devices Illustrated
by Paul Sandin



Both hobbyists and professionals will treasure this unique and distinctive sourcebook — the most thorough — and thoroughly explained — compendium of robot mechanisms and devices ever assembled. Written and illustrated specifically for people fascinated with mobile robots, *Robot Mechanisms and Mechanical Devices Illustrated* offers a one-stop source of everything needed for the mechanical design of state-of-the-art mobile 'bots. **$39.95**

Electronic Gadgets for the Evil Genius
by Robert Iannini



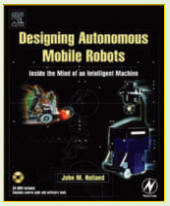
The do-it-yourself hobbyist market — particularly in the area of electronics — is hotter than ever. This book gives the "evil genius" loads of projects to delve into, from an ultrasonic microphone to a body heat detector, all the way to a *Star Wars* Light Saber. This book makes creating these devices fun, inexpensive, and easy. **$24.95**

Robot Builder's Sourcebook
by Gordon McComb

Fascinated by the world of robotics, but don't know how to tap into the incredible amount of information available on the subject? Clueless as to locating specific information on robotics? Want the names, addresses, phone numbers, and websites of companies that can supply the exact part, plan, kit, building material, programming language, operating system, computer system, or publication you've been searching for? Turn to the *Robot Builder's Sourcebook* — a unique clearing-house of information that will open 2,500+ new doors and spark almost as many new ideas. **$24.95**

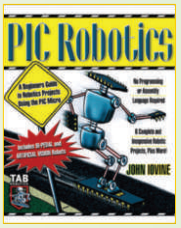
We accept VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.

Designing Autonomous Mobile Robots
by John Holland



Designing Autonomous Mobile Robots introduces the reader to the fundamental concepts of this complex field. The author addresses all the pertinent topics of the electronic hardware and software of mobile robot design, with particular emphasis on the more difficult problems of control, navigation, and sensor interfacing. Its state-of-the-art treatment of core concepts in mobile robotics helps and challenges readers to explore new avenues in this exciting field. The accompanying CD-ROM provides software routines for the examples cited, as well as an electronic version of the text. **$49.95**

PIC Robotics: A Beginner's Guide to Robotics Projects Using the PIC Micro
by John Iovine



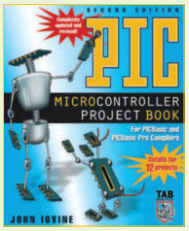
Here's everything the robotics hobbyist needs to harness the power of the PICMicro MCU! In this heavily-illustrated resource, the author provides plans and complete parts lists for 11 easy-to-build robots — each with a PICMicro brain. The expertly written coverage of the PIC Basic Computer makes programming a snap — and lots of fun. **$19.95**

Mobile Robotic Car Design
by Pushkin Kachroo / Patricia Mellodge

This thoughtful guide gives you complete, illustrated plans and instructions for building a 1:10 scale car robot that would cost thousands of dollars if bought off the shelf. But beyond hours of entertainment and satisfaction spent creating and operating an impressive and fun project, *Mobile Robotic Car Design* provides serious insight into the science and art of robotics. Written by robotics experts, this book gives you a solid background in electrical and mechanical theory, and the design savvy to conceptualize, enlarge, and build robotics projects of your own. **$29.95**

To order call 1-800-783-4624 or go to our website at www.servomagazine.com

PIC Microcontroller Project Book
by John Iovine



The PIC microcontroller is enormously popular both in the US and abroad. The first edition of this book was a tremendous success because of that. However, in the four years that have passed since the book was first published, the electronics hobbyist market has become more sophisticated. Many users of the PIC are now comfortable paying the $250.00 price for the Professional version of the PIC Basic (the regular version sells for $100.00). This new edition is fully updated and revised to include detailed directions on using both versions of the microcontroller, with no-nonsense recommendations on which one serves better in different situations. **$29.95**

Robot Builder's Bonanza
by Gordon McComb

Robot Builder's Bonanza is a major revision of the bestselling bible of amateur robot building — packed with the latest in servo motor technology, microcontrolled robots, remote control, LEGO Mindstorms Kits, and other commercial kits. It gives electronics hobbyists fully illustrated plans for 11 complete robots, as well as all-new coverage of Robotix-based robots, LEGO Technic-based robots, Functionoids with LEGO Mindstorms, and location and motorized systems with servo motors. **$24.95**

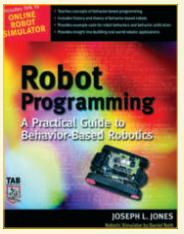
The Ultimate Palm Robot
by Kevin Mukhar / Dave Johnson

Originally developed by Carnegie-Mellon University robotics department graduate students, this prototype has enjoyed a cult following among enthusiasts. Using software provided by the authors and this step-by-step guide, you can build and operate your own version of the same robot. Learn about parts, software, programming, games, robot resources, and much more from this exciting, one stop guide to Palm robots. **$29.99**

Build Your Own Humanoid Robots
by Karl Williams

Build Your Own Humanoid Robots provides step-by-step directions for six exciting projects — each costing less than $300.00. Together, they form the essential ingredients for making your own humanoid robot. **$24.95**

Robot Programming
by Joe Jones / Daniel Roth



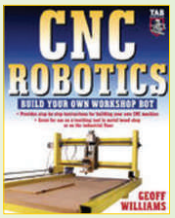
Using an intuitive method, *Robot Programming* deconstructs robot control into simple and distinct behaviors that are easy to program and debug for inexpensive microcontrollers with little memory. Once you've mastered programming your online bot, you can easily adapt your programs for use in physical robots. **$29.95**

Machine Nature: The Coming Age of Bio-Inspired Computing
by Moshe Sipper

Despite being marvels of complexity and human ingenuity, computers are notoriously bad at learning new things and dealing with new situations. Researchers at the frontiers of computer science have turned to nature for solutions to the problem of machine adaptation and learning. By applying models of complex biological systems to the realm of computing machines, they have given rise to a new breed of adaptive software and hardware. In *Machine Nature,* computer scientist Moshe Sipper takes readers on a thrilling journey to the terra nova of computing, to provide a compelling look at cutting-edge computers, robots, and machines now and in the decades ahead. **$24.95**

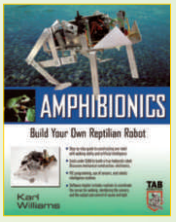
Check out our online bookstore at www.servomagazine.com for a complete listing of all the books that are available.

CNC Robotics
by Geoff Williams



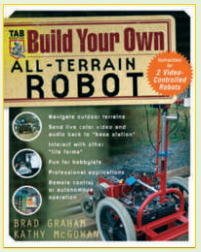
Now, for the first time, you can get complete directions for building a CNC workshop bot for a total cost of around $1,500.00. *CNC Robotics* gives you step-by-step, illustrated directions for designing, constructing, and testing a fully functional CNC robot that saves you 80 percent of the price of an off-the-shelf bot and can be customized to suit your purposes exactly, because you designed it. **$34.95**

Amphibionics
by Karl Williams



If you're a robotics hobbyist with a flair for creativity, here's your opportunity to join the revolution and advance robotic evolution. This work provides the hobbyist with the detailed mechanical, electronic, and PIC microcontroller knowledge needed to build and program snake, frog, turtle, and alligator robots. It focuses on the construction of each robot in detail and then explores the world of slithering, jumping, swimming, and walking robots — and the artificial intelligence needed with these platforms. Packed with insight and a wealth of informative illustrations, *Amphibionics* focuses on construction details and explores the artificial intelligence needed to make these specialized movements happen. **$19.95**

Build Your Own All-Terrain Robot
by Brad Graham / Kathy McGowan



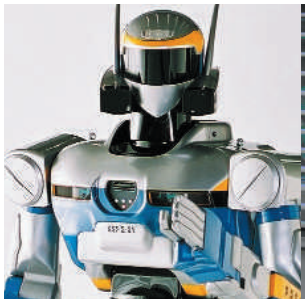
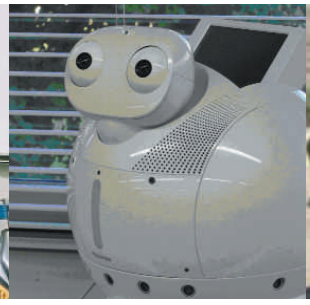
Remotely operated robots are becoming increasingly popular because they allow the operators to explore areas that may not normally be easily accessible. The use of video-controlled technology has sparked a growing public interest not only in hobbyists, but also in the areas of research, space, archeology, deep sea exploration, and even the military. Inside *Build Your Own All-Terrain Robot,* the writers enable even total newcomers to robots to construct a rugged, video-controlled, talking, seeing, interacting explorer bot with a range of over a mile for under $200.00! **$29.95**

www.robonexus.com

produced by:

roboticstrends

IDG WORLD EXPO

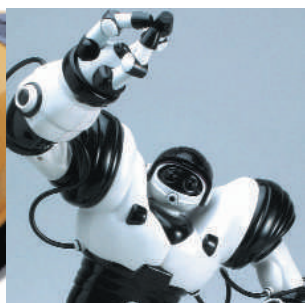



RoboNexus

The Nation's Largest Business Development, Education and Consumer Event for Emerging Robotics Technologies

October 21-23, 2004
Santa Clara Convention Center, Santa Clara, CA

- Business Development and Opportunities Conference
- Emerging Robotics Technologies & Applications Conference
- Robotics Development and Design Conference
- Robotics Education and Instruction Conference
- Consumer Robotics and Entertainment Expo



Premier Co-sponsor

iRobot

Platinum Co-sponsor

evolution robotics

Gold Co-sponsor

HOBBY ENGINEERING

Industry, Media and Association Co-sponsors:

CIRCUIT CELLAR
CoroWare
CYBERBOTICS
therobotics foundry
Industrial Robot
McGraw Hill
robo educators
mBus machineBus.com
NetworkWorld
PC WORLD
ROBOTBOOKS.com
Nuts & Volts
RoboTech CENTER
Robotics Society of America
motioncontrol.com
Entertainment Engineering.com Technology. Creativity. Fun.
SEEGRID Corporation
SERVO
SOFTWARE MAC.COM
WHITE BOX ROBOTICS

Photos courtesy of iRobot, Kawada Industries, Wow Wee Toys, White Box Robotics, Toshiba and ActivMedia Robotics, LLC

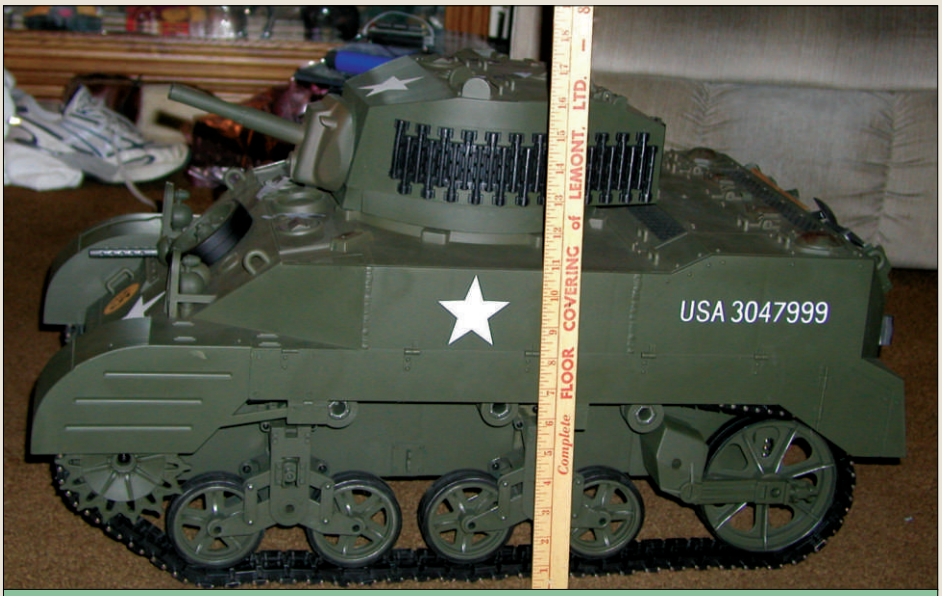
CONVERTING A 1/5TH SCALE M5 STUART TANK

by D. Jay Newman

Wal-Mart sells an R/C M5 Stuart Tank. Well, they sold out shortly before last Christmas. It was sold for under $150.00 and now goes for up to $300.00 on eBay. I'm really hoping that Wal-Mart brings them back.

While they are sold under Wal-Mart's Motorworks label, they were manufactured by 20th Century Toys. Make no mistake; this is an excellent toy for the money. I had fun for a couple of hours just playing with it as an R/C toy in my living room. Its built-in speaker makes a lot of noise, so be careful doing this at night if your spouse is a light sleeper.

When I say it's a good value for the money, that's what I mean. If I wanted a quality 1/5th tank for show or competition, I would spend several thousand dollars and/or several hundred hours making one. This one isn't of that quality. While the treads, suspension, and motors in this tank are of good quality, I don't think they would stand up to too much mud or dirt. (Actually, I think



As you can see, this is a rather large toy and can easily dominate a small house.

that the treads are of excellent quality and are worth the entire price by themselves.) Also, the plastic is not extremely thick or strong. I suspect that this is where they cut costs. Of course, this is one of the reasons why this toy is so good for modification into a robot.

My main goal was trying to create a research platform to use a VIA Mini-ITX motherboard for robotics. I had originally planned on using this as a robotic tank. However, when I tried to fit in the motherboard and an extra battery, I didn't have enough space left. So, I removed the excess plastic that supported the top and just used the base.

The Original Tank

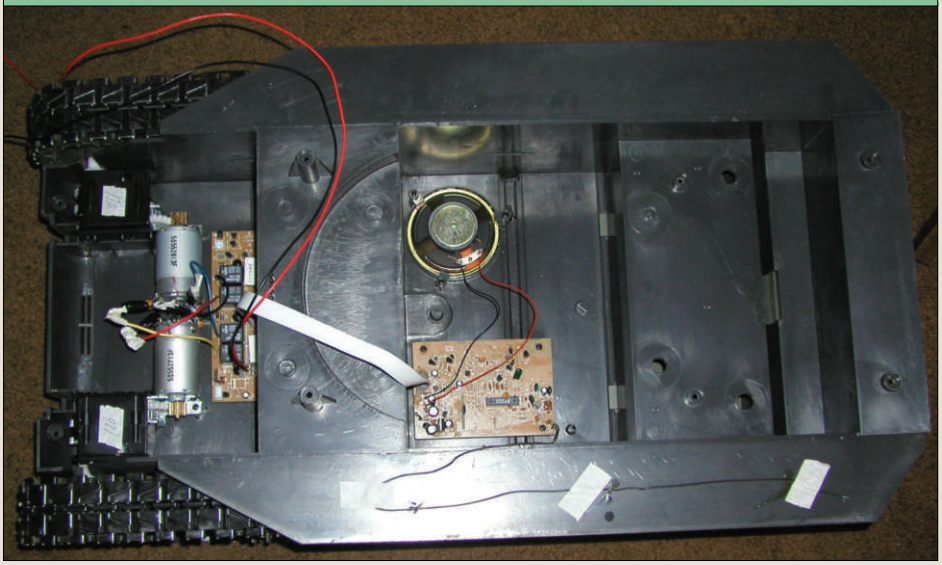
Take a look at pictures of the original tank. Note that there are many compartments. I suspect that the body was built that way to use the compartment sides as supports to make the body stronger.

If you were using a smaller processor — such as a Systronix JStik, a µVM chip, or, indeed, any microcontroller — you would be able to use this tank with very little modification to the body. A microcontroller and its accompanying electronics would fit easily into the main compartment of this tank.

Note: The motors are at the front of the tank.

Unfortunately, I was using a Mini-ITX board, which is around 7.5" square. This didn't fit very easily into the compartment, so I had to make modifications.

The many compartments would house a microcontroller with little modification.



Modifying the Body

Modifying the body is easy.

1. You take off the top. To do this, you must tape all the moveable hatches on the top shut; otherwise, you run the risk of breaking them. Then turn the tank upside down and unscrew everything you can find.

2. Remove all the electronics and wiring, except for the motors.

3. Take a Dremel tool and remove all the internal plastic walls, except the one separating the motor compartment. I left that alone to help make sure I wouldn't mix the motor power and the digital power.

4. Drill three 1/2" holes in the back panel for the battery switches and the motherboard reset switch.

5. Use lots of hot glue to seal the bottom hatches closed.

6. Put a flat base in for your batteries and Mini-ITX board. I cut up some expanded PVC I had around and used that to make a base for an acrylic sheet cut to the dimensions of the inside. The expanded PVC allows the acrylic sheet to be above the not-so-even cuts I made on the plastic.

7. Put some 3M Dual Lock on the acrylic sheet to allow the batteries and the Mini-ITX board to be fastened. I'll admit that I tend to use Dual Lock like most men use duct tape (which, surprisingly enough, I rarely use).

8. Cut another acrylic sheet for the top.

This should be large enough that it fits over the lips of the base. Then, you fasten this down with more Dual Lock, of course.

The Power Supply

The power supply consists of two parts: the batteries and the regulators for the Mini-ITX board and the other electronics.

Batteries

First, there are two 12 volt, 9 amp-hour SLA (sealed lead acid) batteries. The batteries are in the back of the base, side-by-side. Each battery has two sets of wires connected to it; one set goes to the power switch and one set is for the charger. Eventually, I will make a charger that the robot can go to by itself, but — for now — I hook up the charger manually.

The wires are 16 gauge speaker wires and I use 30 amp Anderson Powerpoles for the connectors. I like these connectors because they are very easy to use. I use a $10.00 crimper I bought at Lowes and — after crimping — I use a bit of solder to make sure the connection is good. Then, you put these connectors into their plastic housings, put the housings together, and you have a wonderfully polarized connection.

To keep the batteries in place, I used a bit of Dual Lock on the bottoms. I wouldn't do this on bigger batteries, but it works on these.

I got these batteries over the web at BatteryMart. The price and speed of delivery were excellent.

Each power switch gets the red leads from one of the batteries. These are basic toggle switches I got from Lowes. One controls the motor battery and one controls the electronics battery. Each power switch has one set of input wires and three sets of output wires: two sets of power wires and one set of thinner wires to be used to test the battery charge.

Regulators

I have two regulators on this bot, both connected to the same battery. One controls the Mini-ITX board. This is a Mini-Box PW-70. Just put 12 volts in and you'll get the power supply needed for a Mini-ITX board. This also supplies power to the two compact flash cards used as "hard-drives" by the system. These compact flash HD adapters were also made by Mini-Box.

The only modification I made to the PW-70 was to change the battery connector to a set of Powerpoles.

Now, I have nothing but good things to say about Mini-Box. They have great products. Unfortunately, their distributor of choice seems to be iTuner and — out of the three orders I have placed with them — I have had trouble with two of them.

The Vantec RET 512 solid-state speed controller.

The regulator that controls my electronics is a 5 volt, 3 amp switching regulator from Jameco. I put a couple of caps on it to make sure that the voltage is smooth and put Powerpoles on the power-in connection and the two sets of power-out wires.

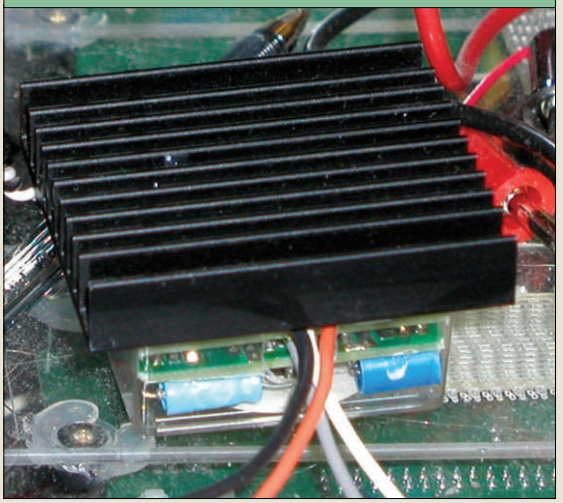
The Mini-ITX Board

As I stated before, I'm using a Mini-ITX board. Specifically, I'm using a VIA M 10000 with 512 MB of RAM running Gentoo Linux. I'll admit that setting this board up was a little challenge; I had to use a real ATX power supply and an internal CD-ROM drive — all sitting on a cardboard box — to install Linux. It can be done purely over the network, but it is much easier to have a CD-ROM drive available. On my next large bot, I will have more power available and get a USB CD-ROM drive or, maybe, I'll do the same thing. Who knows?

Physically, the Mini-ITX board is mounted on an acrylic sheet on standoffs with hot glue. You have to use a lot of hot glue for this to work. Remember to just put a little bit of Dual Lock on the bottom of this sheet. If you put on too much, it will try to hold to the Dual Lock attached to the bottom of the bot too strongly and you'll have difficulty removing the board.

I'm using two compact flash (CF) cards as hard drives. The primary is 1 gigabyte and the secondary is 256 kilobytes. I have the primary setup as "/boot" and "/" and the secondary is set up as "/tmp" and "/var/tmp." Since CF cards are sensitive to the number of times they are written to, I wanted the file systems that were most likely to be

With a heatsink, the RET 512 handles more current.



written to a lot put on a separate card.

Each of these cards is also attached to small acrylic sheets with standoffs and hot glue. These are attached to the motherboard with a standard hard drive cable. I actually used a Belkin round IDE cable. This cable twists and fits into the compact spaces of a bot more easily than a standard cable. The CF hard drives will be attached to the underside of the top panel, again with Dual Lock. I find that Dual Lock fastens these tightly enough to the top so that movement doesn't bother them.

I also mounted two directories — "/usr/src," which is the system source and "/usr/portage," which is the user compiled programs and packages source — via NFS. These directories contain a lot of source code and, hence, aren't needed for the day-to-day operation of the bot. I mount these directories when I want to update my system.

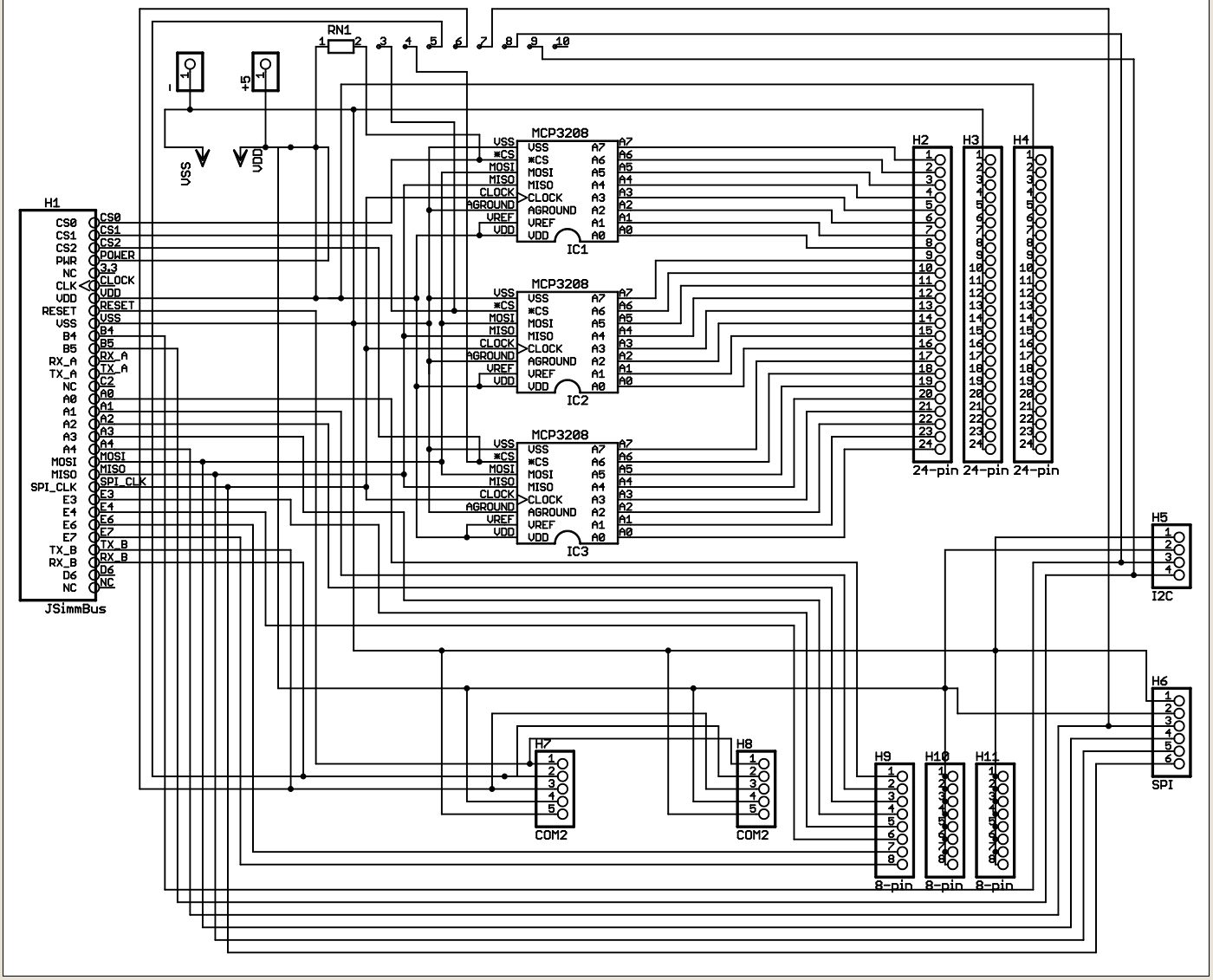
Why did I choose Gentoo Linux?

Sometimes, I think that the choice of an operating system is like a religion or political affiliation. However, I chose Linux because it was free and I chose Gentoo because it was the distribution that was most easily customized to fit a small system. Yes, I know that there are smaller distributions, but Gentoo had what I wanted without requiring me to get everything. I did my research first (I was formerly a Red Hat man) and Gentoo seemed to fit the best. After this, I am installing Gentoo on all my Linux machines — at least until something better comes along.

The Sensor/Motor Controller

As a stop-gap, I am using my JStik and an ADC board (see the August issue of *SERVO*). I plan to eventually create specific motor and sensor con-

The I/O board schematic (for details, refer to my article in the August 2004 issue of *SERVO Magazine*).

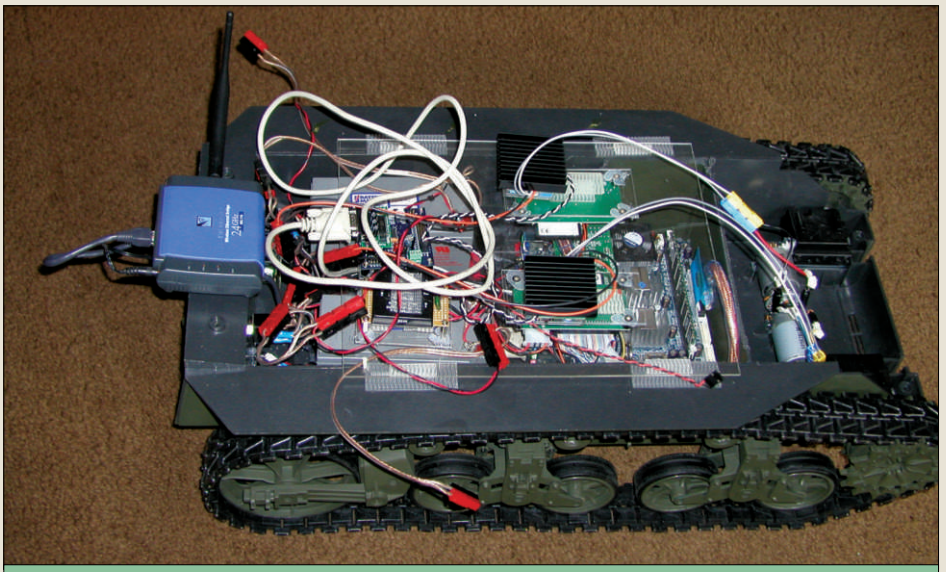


trollers with James Caska's µVM chips. My main objection to the commercial I/O controllers I've seen is that they don't have a proper power structure: I think that power for the sensors should be handled by the controller for most sensors. I would like to see an I/O controller that has both 5 volt and 3.3 volt power rails, as well as signal and ground.

These controllers are connected via the serial port. I use a standard serial cable from my motherboard to the JStik.

The motors are controlled like servos because I'm using two Vantec RET 512 ESCs (electronic speed controllers) to control them. These ESCs are made for radio controlled vehicles, but they work fine for this project. They have battery inputs, R/C servo inputs, and motor outputs. They are controlled just like modified R/C servo motors.

For sensors, my plan is to have eight Sharp IR Rangers, two sonar sensors, several bump sensors, odometry, tilt sensors, a gyroscope, and a compass. This should allow Stuart to have a pretty good idea of where he is.



Here is the completed research-bot. The wires hanging from the sides are there to recharge the batteries and are normally tucked under the platform.

The Software

Most of the low level control software is written in Java and runs on the JStik. I tried to emulate a RoboBRiX-like architecture because I'm thinking of making the final motor and sensor controllers as RoboBRiX. (Why reinvent the wheel?) This software is freely available.

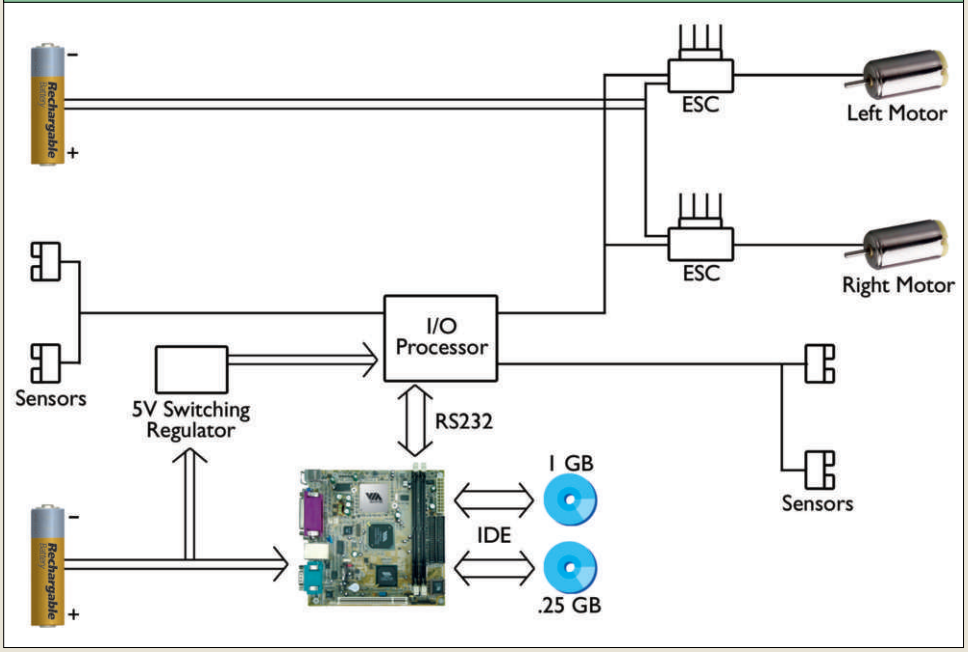
The code to control the robot merely sends commands through the serial port. The command structure is sent as a series of bytes where the first byte is a command and parameters are sent directly afterwards. The command to activate a motor is 0xA0 and the command to set its speed is 0xA1.

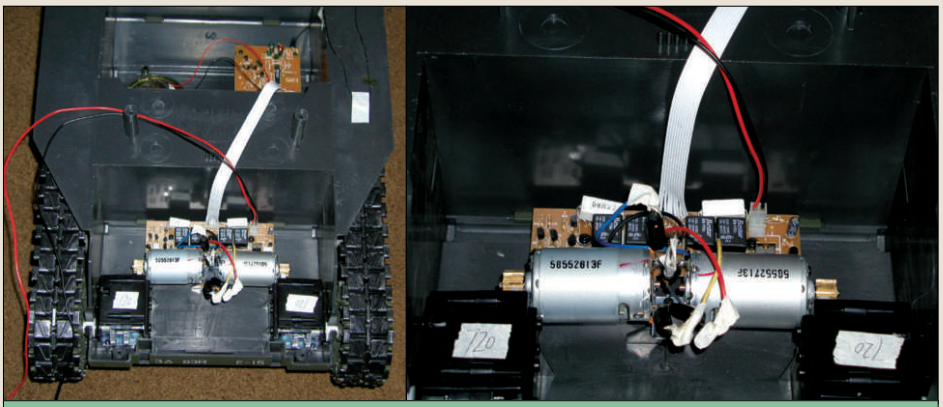
For example, to activate and turn on both motors for two seconds, the Java code is:

```
// Commands to activate the motors
byte[] set0Active =
    {(byte)0xA0, (byte)0x00};
byte[] set1Active =
    {(byte)0xA0, (byte)0x01};

// Commands to turn the motors on
byte[] on0 = {(byte)0xA1,
    (byte)0x00, (byte)75};
byte[] on1 = {(byte)0xA1,
    (byte)0x01, (byte)75};

// Commands to turn the motors off
byte[] off0= {(byte)0xA1,
    (byte)0x00, (byte)0x00};
byte[] off1= {(byte)0xA1,
    (byte)0x01, (byte)0x00};

out.write(set0Active);
out.write(set1Active);

out.write(on0);
out.write(on1);

Thread.sleep(2000);

out.write(off0);
out.write(off1);
```

Currently, the motors are just set

The system block diagram, illustrating both power and control signals.





Detail of the drive motors and original relay controller.

to the requested speed. I am experimenting with ramping the values and using a PID control loop so that the motor speed is set more accurately, and to keep them from burning out with a too sudden speed change.

The sensors are read in a similar way. You activate a sensor and then you can read it. There are a total of 24 ADC ports on this system. The major difference with the sensor code is that reading a sensor forces the I/O processor to send data back. This is sent as two bytes, with the high-byte first.

```
static byte[] activate_0 =
    {(byte)0x01, (byte)0};
static byte[] read_0 =
    {(byte)0x03, (byte)0};

out.write(activate_0);

// Check 10 times per second
while (true)  {
  Thread.sleep(100);
  int value = readSensor(read_0);
}

static public int readSensor(byte[]
command)  throws IOException  {
   out.write(command);

    byte[] buffer = new byte[2];
   in.read(buffer);

    int value = buffer[0] & 0xFF;
    value <<= 8;
    value |= (buffer[1] & 0xFF);

    return value;
}
```

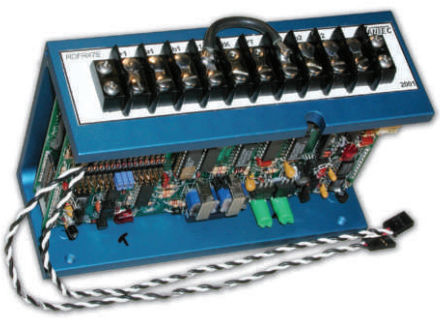
Network Connectivity

Linux is good at TCP/IP networking and the VIA motherboard I'm using has an Ethernet port built in, as well as a PCI slot.

I got a good deal on a Linksys 802.11b Ethernet to WiFi bridge. This is just a device with a single Ethernet connection, a WiFi section, and some circuitry to bridge the two. Setup is easy and it takes 5 VDC. This is the second device powered by the Jameco 5 volt power supply. Basically, I need to give the Mini-ITX board and the bridge ip numbers. It really is that easy.

I'll admit that WiFi doesn't have the range that some of

STEER WINNING ROBOTS WITHOUT SERVOS!



Perform proportional speed, direction, and steering with only two Radio/Control channels for vehicles using two separate brush-type electric motors mounted right and left with our **mixing RDFR dual speed control**. Used in many successful competitive robots. Single joystick operation: up goes straight ahead, down is reverse. Pure right or left twirls vehicle as motors turn opposite directions. In between stick positions completely proportional. Plugs in like a servo to your Futaba, JR, Hitec, or similar radio. Compatible with gyro steering stabilization. Various volt and amp sizes available. The RDFR47E 55V 75A per motor unit pictured above. **www.vantec.com**

VANTEC

Order at
(888) 929-5055

RESOURCES

VIA — Small, low power motherboards
www.viamainboard.com/

BatteryMart — All sorts of batteries:
www.batterymart.com/

Vantec — High performance speed controllers for robots
www.batterymart.com/

Mini-Box — Power supplies and compact flash to hard drive converters:
www.mini-box.com/

My site — Home of my Java behavioral programming framework:
http://enerd.ws/robots/code/

the wireless modems have, but it makes it trivial to put my bot on the network.

The Camera

I got a Logitech Orbit webcam. This is a neat camera that can be panned and tilted via a Linux driver. It has a stalk that can be used to raise the camera up about eight inches. I am just starting to hook this camera up and get control and pictures from it. This doesn't require a separate power supply because it uses the USB port to power it.

I will admit that the camera gives me strange ideas. I want to use the stalk and have a motor move the camera up and down. Imagine, if you will, the movements and sound effects:

Camera moves up ("zzzzork").

Camera quickly pans left ("zzzz-wwwweep").

Camera quickly pans right ("zzzz-wwwweeeep").

Camera quickly goes back down ("zzwwwoooop").

Okay, I'll admit to having a juvenile sense of humor sometimes.

Future Considerations

Eventually, I would like to put the top back on. Not only would Stuart look more like a tank again, but the turret would be a nice place to mount another camera and a laser. Every bot should have a laser. Even another sonar unit attached to the turret would make things easier.

Another consideration for the future is to use the PCI 302.11g card I have. Right now, there is no Linux driver, but there should be one, shortly.

I also have a behavioral programming framework that there isn't enough time to talk about right now, but it is downloadable from my site. SV

ABOUT THE AUTHOR

During the day, D. Jay Newman works with a group called TLT at Penn State. In the evenings and on weekends, he is busy constructing robots, writing, and — when he has the spare time — goofing off. He is currently writing a book called *Linux Robotics: Building a Smarter Robot*, to be released onto unsuspecting bookstores sometime in 2005. He lives with a wife, a dog, and an annoying parrot. He enjoys writing about himself in the third person.

Robotics Showcase

ALL ELECTRONICS CORPORATION
THOUSANDS OF ELECTRONIC PARTS AND SUPPLIES
VISIT OUR ONLINE STORE AT www.allelectronics.com
WALL TRANSFORMERS, ALARMS, FUSES, CABLE TIES, RELAYS, OPTO ELECTRONICS, KNOBS, VIDEO ACCESSORIES, SIRENS, SOLDER ACCESSORIES, MOTORS, DIODES, HEAT SINKS, CAPACITORS, CHOKES, TOOLS, FASTENERS, TERMINAL STRIPS, CRIMP CONNECTORS, L.E.D.S., DISPLAYS, FANS, BREAD-BOARDS, RESISTORS, SOLAR CELLS, BUZZERS, BATTERIES, MAGNETS, CAMERAS, DC-DC CONVERTERS, HEADPHONES, LAMPS, PANEL METERS, SWITCHES, SPEAKERS, PELTIER DEVICES, and much more....
ORDER TOLL FREE 1-800-826-5432
Ask for our FREE 96 page catalog

Surplus Sales of Nebraska
www.surplussales.com
NVidia Graphics Card Cooling Fan
(FAN) MTHERM-1027B
NVidia graphics card cooling fan fits NVidia VisionTek. 3.26 cfm, 12 vdc, .09 amps. 2 pin connector.
Dimensions: 68mm x 60mm x 10mm high. Manufactured by Millennium. 26,000 are available!!

$5 each - $3.75 (12+), $3.25 (100+), $2.75 (1000+)
800-244-4567
402-346-4750
visit our website @
www.surplussales.com

BEAM Robotics Step by Step PART 2

An Intro to BEAM and the BBPV

by Thomas Gray and J. Wolfgang Goerlich

Last month, we introduced BEAM (Biology, Electronics, Aesthetics, and Mechanics), which originated with Mark Tilden — whose recently released RoboSapien was featured in the May and June issues of *SERVO* — as a way to create simple robots that can mimic basic biological behaviors. We presented a little light-seeking robot called the Bare Bones Photovore that used a single inverter IC. However, Tilden's conception of BEAM uses artificial electronic neurons linked in Nervous Nets. In this article, we'll move toward slightly more complex BEAM robots.

The first step toward building a Nervous Net is to use an inverter in what Tilden termed a "Nervous Neuron" or Nv Neuron (Figure 1). The Nv Neuron only has three parts: a capacitor, a resistor, and an inverter. The junction of these three components is the bias point and will be important down the road. Meanwhile, let's start by using the Nv together with switches and other basic sensors.

The Nv Neuron

The Nv Neuron mimics a biological neuron in that it becomes active when an input signal comes in and then remains active for a period of time. The Nv is essentially a timer. A signal comes in, the inverter turns on, and it flips the signal to the output channel for a specified amount of time.

Now, it's time to breadboard. Plug in the 74AC240 with its power line (pin 20) and ground lines (pins 10, 1, and 19). We are going to use the first inverter in bank one. It goes from the input at pin 2 — labeled 1A1 — to the output at pin 18 (1Y1). For convenience, we can identify that particular inverter as Inv(2,18) and a Nv Neuron built on that inverter as Nv(2,18).

Figure 2 shows the 4.7M resistor connected from pin 1 to ground, while the 0.22 µF capacitor is connected from pin 1 to a row of holes above the IC. That row of holes will connect to our input.

You'll also see that each unused input is tied to Gnd. Inputs that are left unattached tend to wander and get into trouble, so you have to fasten them down to either Vcc or Gnd. This is one of the things that everyone assumes you know, so no one ever tells or shows you this in circuit diagrams.

We need a way to see the output. Figure 2 also shows a resistor (470 Ω to 1K) from pin 18 to a row of unoccupied holes on the other side and an LED from there to Vcc. The LED cathode (the flat side or the short lead) is toward the inverter output. The resistor limits the current flowing through the LED and inverter so that nothing burns out. The circuit is shown in Figure 3.

Figure 1. A single Nv Neuron.

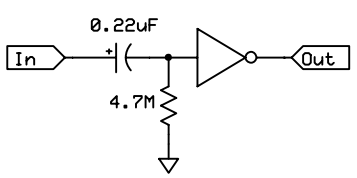


Figure 3. Nv Neuron with process indicator.

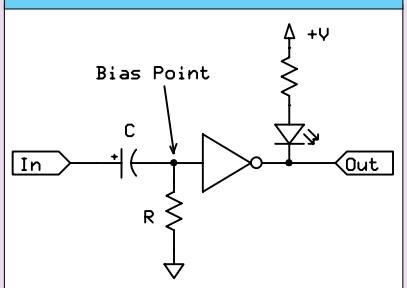
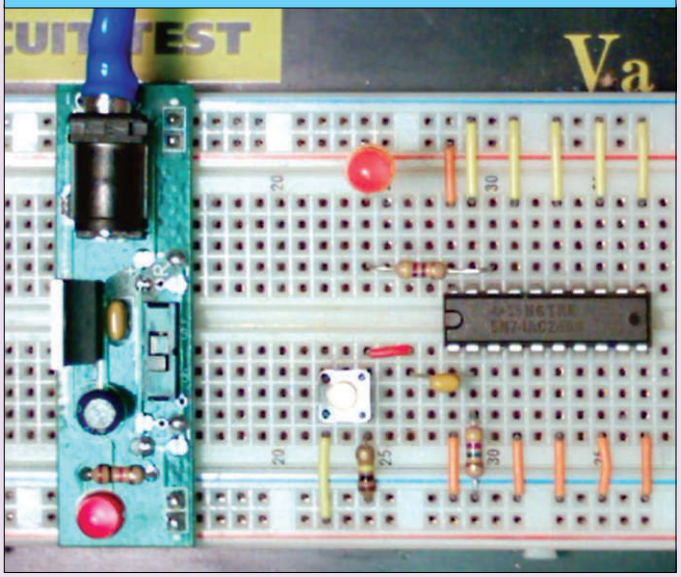


Figure 2. Breadboard 2 shows the resistor, capacitor, and inverter, along with the output resistor and LED.



Connect a tie from the input row to Vcc. The LED should glow briefly and go out. If it does — congratulations! You've built an Nv Neuron with an on time of about a second. The entire transition — from low to high and then to off — is called a process.

Want to see it again? Remove the tie from input to Vcc and reconnect it. Nothing. Briefly connect the input to ground (bring it low), then reconnect it to Vcc (bring it back high). Oops, there — the LED flashed. This makes

sense. You have to turn a switch off (low) before you can turn it on (high) again. This resets the Nv Neuron. The in between state — neither off nor on — is a source of problems and, therefore, inputs have to be pulled high or low.

Before we go further, let's add a process initialization circuit (PIC) to make it easier to activate our Nv Neuron. A pushbutton and a 100K resistor will work. onnect the switch's common pin (marked C, if you have that type of switch) to the Vcc and the normally open pin (marked NO or maybe not marked) to the resistor. Hook the parts up as illustrated in Figure 4.

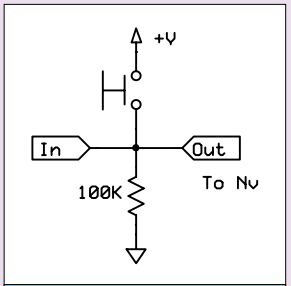


Figure 4. Process Initialization Circuit (PIC).

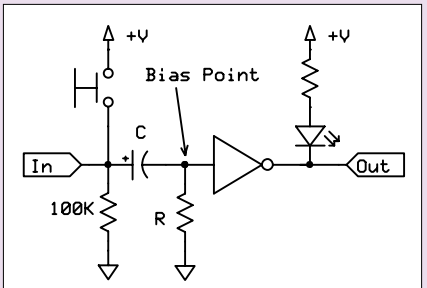


Figure 5. Nv with indicator and PIC.

The 100K resistor normally pulls the input low; for this reason, it is called a "pull-down resistor." With a low input, the output is always inverted to high, which — in this case — means "inactive."

Pushing the button bypasses the pull-down resistor, quickly charges the Nv's capacitor, and brings the input high. The Nv's inverter switches the input to low and the Nv is active. Low = Ground = Battery Negative, so a circuit that is On in the Low state is said to follow Negative Logic. It makes sense when you think about it.

When you power up, the LED should be off. Pushing the button makes it go on briefly. When the voltage in the capacitor is drained by R1 to where it falls below the switching point (1/2 of Vcc for the 74AC240, remember?), the inverter switches back to high, the LED turns off, and the Nv is said to be inactive.

Here's what we've built so far, all combined into one wonderfully complete diagram (Figure 5).

The Nv Neuron stays active for a time period determined by the values of the resistor and capacitor and then automatically goes off, even if the input is still connected. You can hold that button down for as long as you want to, but the output pulse is going to blink off at its predetermined time.

Determining the Delay Time

Put simply, the bigger the capacitor or the greater the resistance, the longer the Nv stays on. Electronics gurus call this combination the "RC Constant." In math terms,

T = R • C

The time in seconds equals the resistance in megohms times the capacitance in microfarads.

Suppose then that you have a 4.7M resistor and a 22 µF capacitor:

T = R • C
= 4.7 • 22
= 103.4 seconds

You can estimate this easily in your mind: roughly 5 multiplied by roughly 20 is about 100). For most BEAM purposes, the rough estimate of 100 seconds — just over a minute and a half — is a good enough answer.

Anyway, if you need to have a Nv Neuron active for about 5 seconds, you can use:

T = 4.7 µF • 1M or
= 2.2 µF • 2.5M

BUT WAIT!

Attentive readers may have noticed a logical flaw with this formula. The problem is that it assumes that the capacitor is completely drained before the inverter switches. When we looked at the BBPV, we saw that 1/2 of Vcc is the switching point for the 74AC240 and 74HC240. So, your timing will be close, but not exact.

Figure 6. Nv Neuron with On/Off LEDs.

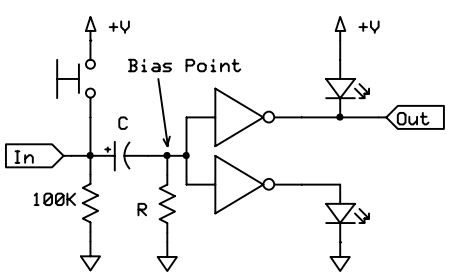
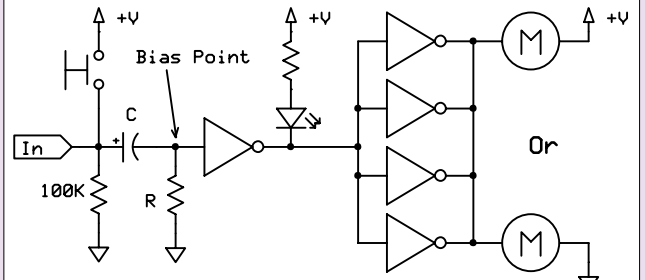


Figure 7. Nv TimerBot.



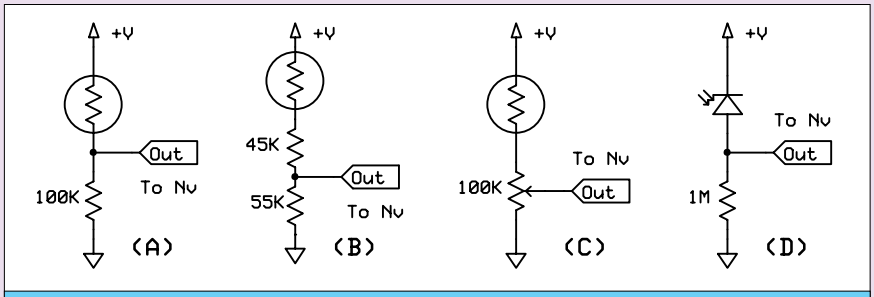


Figure 8. Light activated PIC and Dark Lurker. **(A)** CdS cell and 100K pull-down. **(B)** CdS cell with 45K series resistor and 55K pull-down. **(C)** CdS cell with 100K trimpot. **(D)** Photodiode with 1M resistor.

= 1 µF • 4.7M,
= 0.47 µF • 10M, and so on.

Any of these combinations will give you approximately 5 seconds.

Back to Figure 5. The Input and Output flow boxes show where you can connect the Nv Neuron to something else. When the inverter turns on, the output signal can be sent to another circuit to turn it on or off. This other circuit might be a motor driver, a sound maker, a relay, or even another neuron. We'll hold off on chains of neurons until later. For now, here are some other ideas for a single Nv Neuron. Remember, the Nv is a timer that says, "When you get this input, activate the output for a specified time."

Light Show: Add an "Off" LED of a different color, hooked through a resistor to ground, through Inv(4,16) as shown (Figure 6). In this case, we have two output devices: one showing the "Off" state and one showing the "On" state. (Perhaps BEAM purists will point out that we no longer have a single Nv, but the blinky lights are fun and help show what is going on.)

We still have our output block showing that we can still connect this Nv to something else. Also, we can use the "Off" state to control one device and the "On" state to control another device, such as a motor.

TimerBot: We can make a motor neuron by connecting the Nv Neuron to the four inverters on the other side of the 74AC240. As shown in Figure 7, the Nv will spin the motor when active with the motor wired to ground. What do you think the alternative — with the motor hooked to positive — will do? If you have two motors, try one each way and see what you get.

We almost have a robot here, although it is one with a tiny brain. Let's add sensors to create some simple robots that respond to external stimuli.

Dark Lurking Bot: First, we'll give our robot the gift of sight by replacing the PIC switch with a CdS photoresistor (Figure 8). Like the Bare Bones Photovore's reverse-biased photodiodes in last month's article, the pull-down resistor and the CdS cell form a voltage divider. Its output is used to influence the Nv.

The Solarbotics CdS used in our sample bot has a light resistance of less than 1K, and a dark resistance of about 8K. To prevent excessive current through the divider, we'll want another resistor with the photo resistor (Figure 8b). Replace the pull-down with a 100K potentiometer for more control (Figure 8c).

You might also want to pull one of the

PARTS LIST

Electronics		
74AC240*	Texas Instruments SN74AC240N	Mouser 595-SN74AC240N
	Fairchild MM74AC240N	Solarbotics 74AC240
Photodiodes	Siemens SFH 205f wide field	Solarbotics IR1
	Salvaged from computer mouse	
Assorted LEDs*	Tiny red LEDs	Mouser 638-204HD
	Tiny green LEDs	Mouser 638-204GT
	Tiny red, green, or yellow LEDs	Solarbotics TLED
Resistors	Assorted, a few 470 ohms — 1K *	
	And a few in the 1M-10M range	
Capacitors*	Assorted, 0.1 µF-0.22 µF range	
Remocon Sensor*	PNA4602 or equivalent	Solarbotics PNA4602
CdS Photoresistor	Most types should work well	Solarbotics CDS
Mechanics		
Motors*	Two matched hobby gear motors with wheels	Solarbotics GM6 with RW2
	Two recycled matched “pancake” motors	CD players or CD-ROM
5V Power	Four-AAA or four-AA battery pack*	RadioShack 27-411 (includes power switch)
Power Switch	SPDT or SPST power switch*	Various suppliers
Misc.		
Breadboard*	Generic solderless breadboard and ties	Various suppliers

** Required for this month's projects*

photodiodes off your BBPV and try that with a 1M resistor for a miniscule lurk current (Figure 8d).

In the dark, the Nv is held low (off). When light hits the CdS cell (or PD), the voltage goes up, the Nv becomes active, and your robot will scoot forward. (You might want to change the capacitor and resistor values in your Nv to give it the right amount of motion.) This creates an "attack bot" that lurks in the dark and scoots forward when light hits it. It's kind of cute and — if you tie some lights and a buzzer in with the motor output — it's something with a fair bit of startle value.

This bot is a bit reminiscent of the hunting spiders that lurk in burrows and spring out when insects wander past.

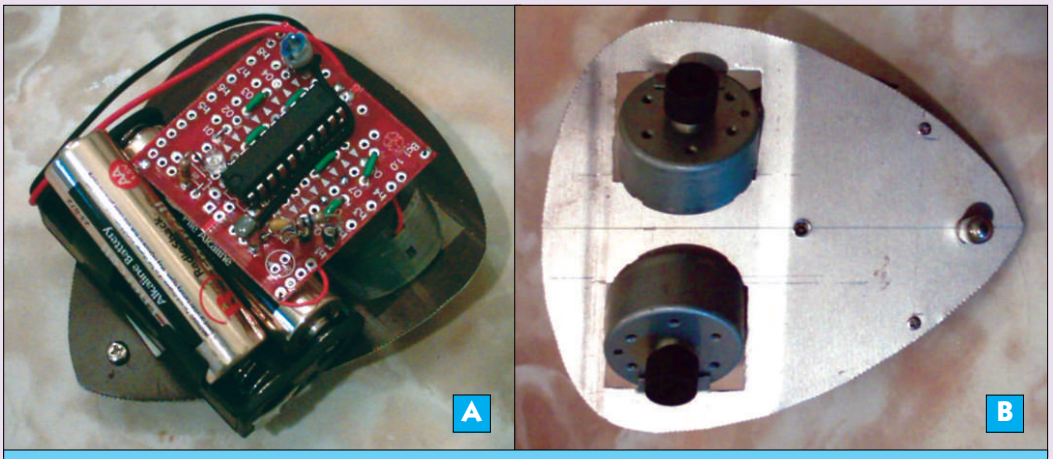


Figure 9. The Dark Lurker, using a 3 V battery pack and a PCB from the Solarbotics BEP kit. The motors are salvaged from audio cassette players. Lurking only consumes a few mA, so it can hide in the dark for a while. **(A)** Top **(B)** Bottom

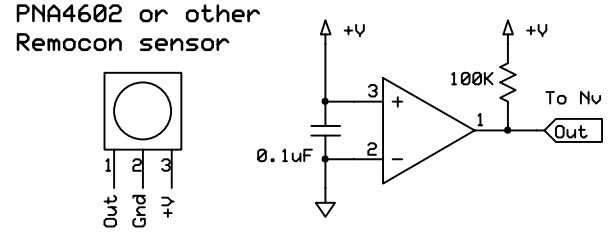
Remote Activated Bot: Finally, let's take a quick look at remote controlled PICs. Typical Remocon sensors — such as the PNA4602 — have an output pin that goes low when the 38 kHz remote carrier wave signal comes in. The output then beats to the drum of the remote. If the remote doesn't send any data, then PNA4602 remains low as long as there is a signal. Try the PNA4602 PIC circuit shown in Figure 10.

When there is no signal, the PIC output to the Nv is high. The Nv fires once when powered up and then stays inactive. If just a 38 kHz signal comes in — say, from the bow of a Sumovore — then the PNA4602's output goes low, resetting the Nv. The Nv fires when the signal stops.

Now, try using a television remote control. The remote sends a 38 kHz signal *and* data on that signal. The PNA4602's output beats with the data, resetting and firing the Nv. When you push the button, the bot moves.

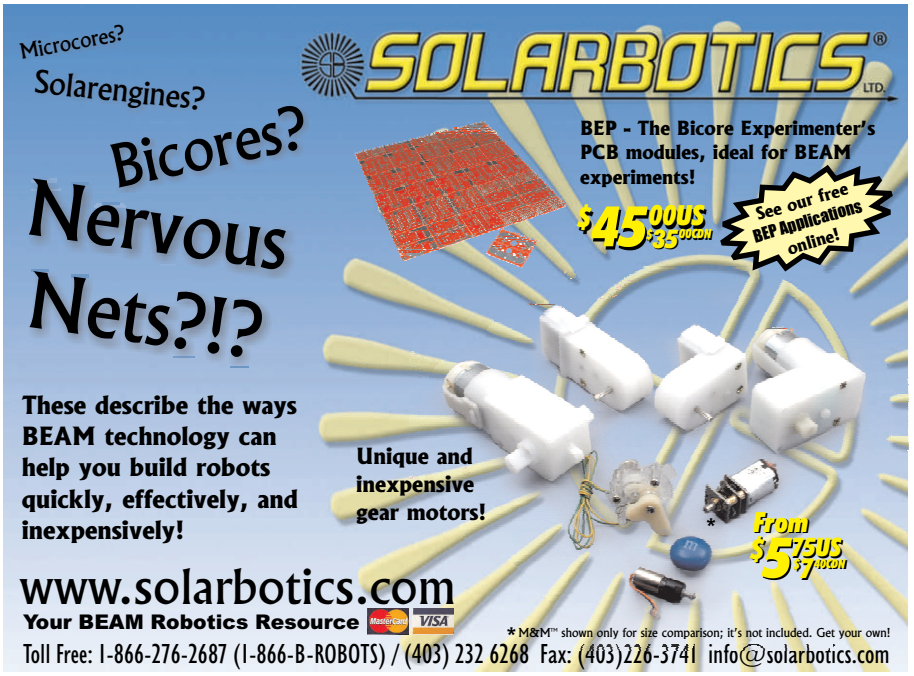
Okay, so we're a little bit away from, "Push channel up button and the bot moves left; push channel down button and the bot turns right," but it's a start! Have fun exploring. SV

Figure 10. Remocon PIC for the TimerBot.



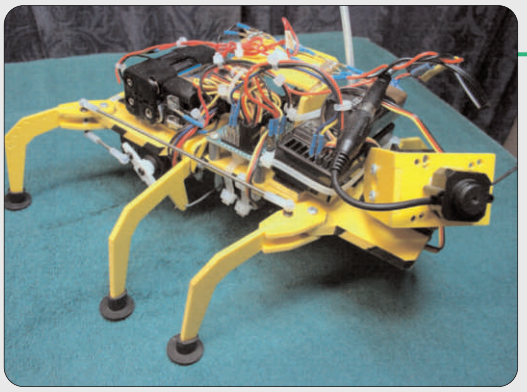
ABOUT THE AUTHORS

Tom is an R & D Communications Officer for a company that manufactures hot tubs. Wolfgang works in IT and specializes in thin-computing solutions. Hobbyists with absolutely no formal technical or electronics background whatsoever, the two connected online at ***http://groups.yahoo.com/group/beam/*** *and have never met face-to-face.*



MENAGERIE

Send us a high-res picture of your robot with a few descriptive sentences and we'll make you famous. Well, mostly. menagerie@servomagazine.com



Col. Hogan

Dave Sweet, Upstate, NY

This is my first generation model of a remotely piloted hexapod with alternating tripod gait. It uses a BASIC Stamp I to interpret R/C signals, and convert them to subroutine mobility commands with full omnidirectional control. It also sports an independant tilt and pan wireless camera. The chassis was provided by Lynxmotion and is made of laser cut polystyrene. Many electrical connections are routed through an open custom-built terminal for easy reconfiguration, and test bed applications. Previously, the Colonel used a 12 VDC system for the camera and transmitter, with a voltage divider to 6 VDC to power the receiver and R/C servo bus. An upgrade to a 9 VDC camera system eliminated over a pound of batteries.

Jx3P@yahoo.com

Scraps and GPSbot

Dan Pickard, Spokane, WA

I built my first robot "Scraps" (Guess how it got its name?) around a PIC microcontroller. It's pretty basic, really. It uses R/C servos attached to model aircraft wheels for mobility. A 7.2 VDC R/C car battery sits under its belly. On the top, an H-bridge, key pad, and LCD all communicate on one data bus to Port B of the PIC. (I used Port A with a few AND gates as a multiplexer for data control.) The PIC was programmed with a simple language that allows you to write a control and duration sequence to the EEPROM. You then hit the start key and away it goes!

My second project is the shizznit! I built a high performance outdoor and in-the-dirt robot base unit for only $30.00. I used $5.00 10" pneumatic tires from Harbor Freight and secondhand $6.00 windshield wiper motors from the junkyard. Each motor pulls 2A at 12 VDC and 9A at stall. I get "walking speed" at 12V. It is fast enough to be entertaining. Future plans are to add an Oncore GPS, up the motor voltage to 24V, and add more microcontrollers! I think it could violate FCC rules, but I'm going to try to build a simple PIC-based modem and use the VOX personal two-way radios for a remote data link to move RS-232. Remote video would be cool also. Hmmmm ...

dpickard@servershare.com

Follow that Line

Adding a Line Tracker Circuit to the OctoBot Survivor™ Robot

by Roger G. Gilbertson

Humans and Robots — the Early Days

Robot groups provide their members with exposure to new ideas, new robotic problems and solutions, and — most importantly — an essential way to meet like-minded robot builders and wanna-be builders. These social benefits cannot be overvalued, as they provide face-to-face interaction with other intellectually compatible humans — something that many of us who have strong "nerd" tendencies might otherwise go without.

When I first got involved with hobby robotics back in the early 1980s, I joined both of the two local robotics groups, each with its own distinctive features and benefits.

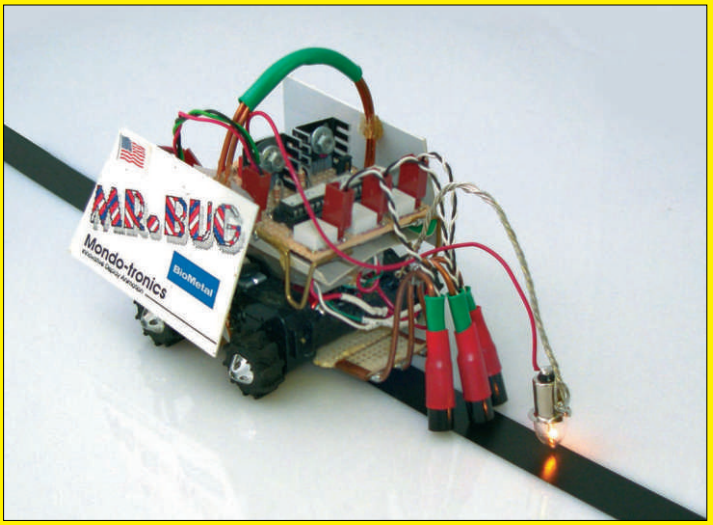
First, I discovered the HomeBrew Robotics Club (HBRC), which — for many years — held monthly meetings in Palo Alto, CA. Mr. Richard Prather championed this group, which formed as an outgrowth of the HomeBrew Computer Club (which featured the early appearance of the Apple 1 computer kit, produced by young Steve Wozniak and Steve Jobs). Dick announced the regular monthly meeting via a simple photo-copied mailing (membership dues just covered the cost of postage) and he held well-focused and very productive evening meetings that provided everyone with regular robotics recharges in those "pre Internet" days.

Second, I joined the San Francisco Robotics Society of America (RSA), which met regularly for many years at San Francisco's innovative science museum — the Exploratorium.

Though the leadership of this group changed more often than the Palo Alto group, SFRSA held yearly robotic competition events that provided me with a good goal for actually getting a robot built.

Remember that, before the advent of "Robot Wars" (the very first of which I attended with the then-SFRSA president Brad Smallridge — it blew us

Figure 1. Mr. Bug, a line following robot that uses a simple "cross over" circuit without a microprocessor.



BASIC Stamp is a trademark of Parallax, Inc. OctoBot Survivor is a trademark of Mondo-tronics, Inc.

Figure 2. OctoBot Survivor Robot with line tracking circuit.

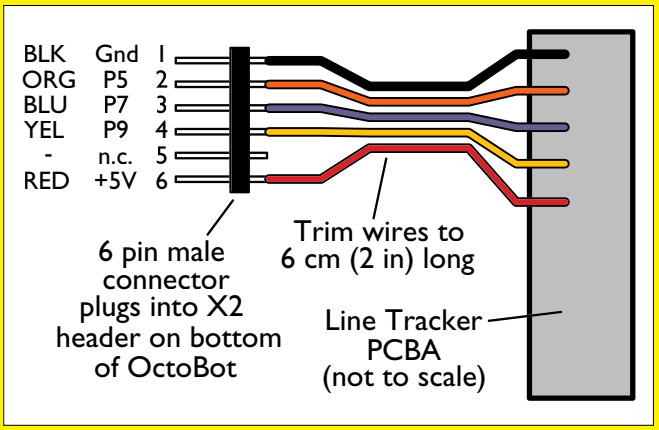


Figure 3. Wire connections.

both away!), robot competitions mostly involved simple maze challenges, line following events, speed demonstrations, and "open"competitions, where entering robots could demonstrate whatever they did best.

Though they lacked the "rock and roll" of many modern robot events, these gatherings kept us robot builders going throughout the year.

The first robot I ever built for a competition was for just such an RSA event, held on a rainy afternoon in the community meeting room of a local bank.

Though not terribly fast, my line follower — "Mr. Bug" — proved that reasonable accuracy combined with reasonable speed gave enough of an edge to win (at least that year) and we received a blue ribbon for our efforts.

Mr. Bug used a single light emitter (a flashlight bulb), three light sensors (photocells), and a simple "cross over" network circuit that caused the right motor to run if the left sensor saw the line, the left motor to run if the right sensor saw the line, and both motors to run if the center sensor saw the line. Such a simple approach works great with simple line paths, but not as well with branching paths, complex intersections, lost lines, dead ends, etc.

For "logical" solutions to more complex problems (line following and others), the addition of a microprocessor robot brain opens up whole new worlds of detection, decision, and control.

Line Following in the New Millennia

Now — years later — with low cost and amazingly powerful microcomputers employed nearly everywhere, robot builders still find themselves faced with line following tasks for competition and other reasons.

Plus, line tracking challenges are simple enough to get beginners interested and involved, but complex enough to keep more seasoned builders scratching their heads at the finer points of the challenges.

Starting in the fall of 2002, I began developing the OctoBot Survivor Robot kit with my team of skilled robot builders (many of whom I first met via the local robot clubs). We designed the OctoBot as a self-charging robot kit for experienced robot builders and included enough expansion possibilities so that students, hobbyists, and experimenters could use it as a platform for testing the boundaries of self-charging robotics and other robot functions.

The OctoBot features two expansion headers (one on top and and a smaller one on the bottom side), as well as a socket for a Stamp 2 type microcomputer. This project turns the OctoBot into a great line tracking robot platform. It shows how to add a secondary "über brain" to the OctoBot for taking control of its built-in PIC876 brain.

For more on the OctoBot Survivor and the challenges of "always on" robotics, see the November 2003 issue of *SERVO Magazine*.

Let your OctoBot Survivor Robot

Quantity	Description	RobotStore.com Item #
1	OctoBot Survivor Robot kit	3-918
1	OctoBot Stamp 2 set (with Stamp 2 and serial cable)	3-905
1	Line tracking sensor (assembled and tested)	3-554
1	Header, six pin male, 0.100"spacing	
3	Spacer, hex, aluminum, 1/2" long, female, #4 thread	
5	Screws, #4, 1/4" long, pan head	
1	Support bar, aluminum, or PC board material	

Parts List

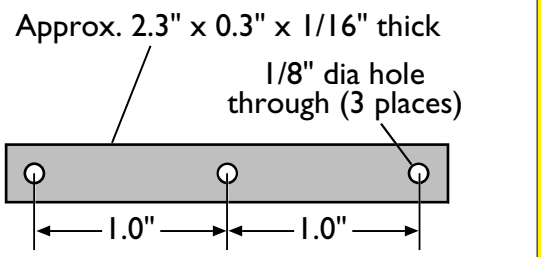


Figure 4. Fabricate the support bar.

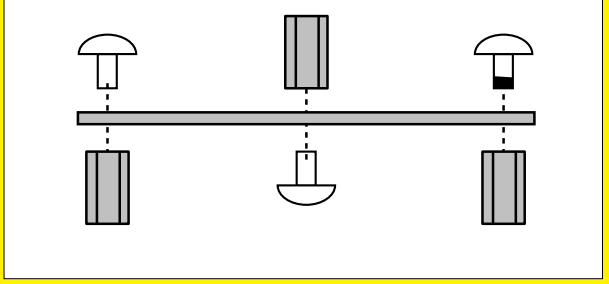


Figure 5. Assemble the support bar and spacers.

detect and follow a line trail wherever it may lead by adding a Lynxmotion Line Tracker sensor, which comes fully assembled and tested, and a Parallax Stamp 2 processor. The accompanying notes include everything you need to add the hardware and software to your OctoBot. With a bit of soldering and a dash of programming, your OctoBot will be moving on down the line!

In addition, you will need a fine tip soldering iron, solder for electronics, a moist sponge (to keep the soldering iron tip clean), side cutters, wire stripper, a PC with a serial port, and Stamp 2 programming software. You can download the Stamp 2 programming software, the Stamp 2 manual, and the OctoBot line tracking demonstration program — OctoLine1.BS2 — either from the *SERVO Magazine* website (**www.servomagazine.com**) or from **www.RobotStore.com/support.asp**

Assemble the parts as described below, plug a Stamp 2 into your OctoBot, connect to your PC, run the Stamp 2 programming software, and program your robot with the demonstration program.

Preparing the Parts

1. With wire cutters, shorten the five wires from the Line Tracker PCBA to about 6 cm (2 inches) in length. Then strip the insulation from the end of each wire by about 3 mm (1/8 inch) and tin the ends of each wire with solder. Then, solder the wires to the six-pin male connector, as shown in Figure 3. Note the color and location of each wire and be sure to leave pin 5 unconnected. When you're done, double check all connections.

2. Fabricate the support bar from thin metal, plastic, or printed circuit board material. Cut the rectangle to size and drill three holes, 3 mm (1/8") in diameter, as shown in Figure 4.

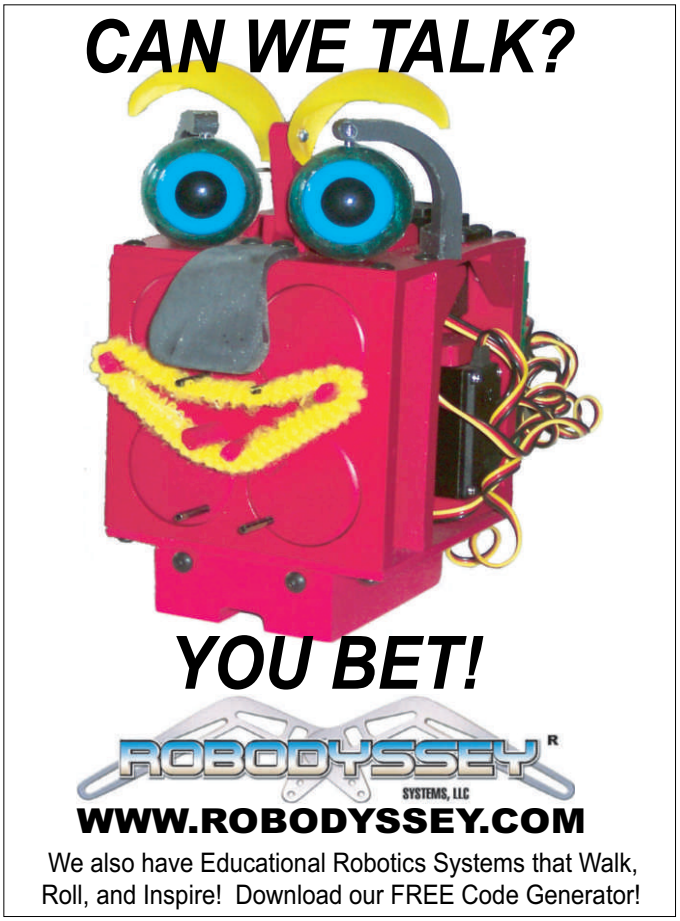
3. Mount three spacers to the support bar using screws, as shown in Figure 5.

Installing the Items

1. Remove the forward nut from the bottom side of OctoBot; you may need to remove a battery from the front battery holder. Attach the support bar with its spacers to the screw, as shown in Figure 6.

2. Plug in the six pin connector into the X2 header on the bottom of the OctoBot (double-check its orientation!) and secure the line tracker board to the two standoffs with the line tracking sensors facing downward and positioned toward the front of the robot, as shown in Figures 7 and 8.

3. Remove the top shell from the OctoBot, install the Stamp



Circle #86 on the Reader Service Card.

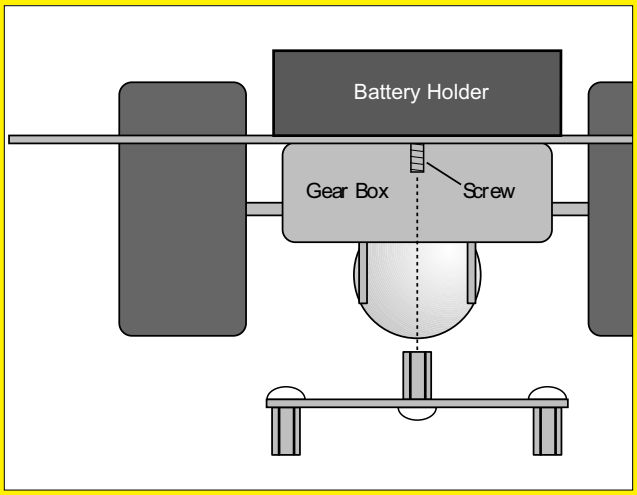


Figure 6. Front view of OctoBot (some details omitted for clarity). Mount the support bar to the bottom screw on the robot.

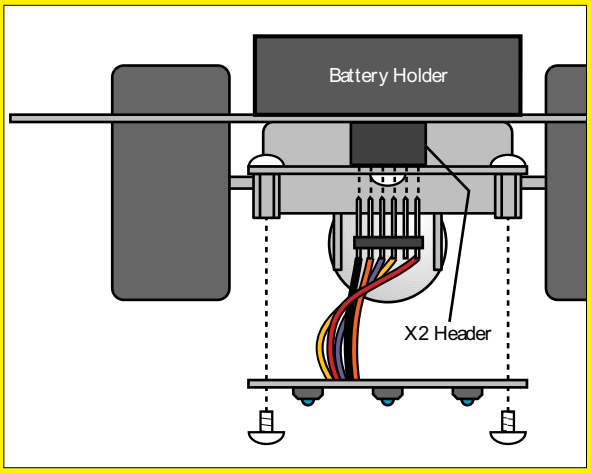


Figure 7. Mount the line tracker to the support bar and plug into the X2 header.

2 into socket U11 (note proper orientation!), plug the serial cable into header J2 (again with proper orientation), as shown in Figure 9.

4. Finally, launch the Stamp 2 programming software on your PC and download the OctoLine1.BS2 demonstration program to try out the line tracking sensor system.

To start with, try running a length of black electrical tape on a large sheet of white paper or cardboard. You can expand your maze to make a track — complete with curves, turns, and crossings. You may need to expand on the program below to help it handle "special features" in your maze.

You can also make a white line on black maze and make appropriate modifications to the program. Explore!

Resources

Lynxmotion — **www.lynxmotion.com**
Parallax — **www.parallax.com**
Mondo-tronics — **www.RobotStore.com**

Program Features

After you've downloaded the program, take a look through it to see how it operates. Then, use it — together with the Stamp 2 manual — as starting points for your own programs and routines.

From here, you can experiment with your own changes and additions to the line following program. Plus, you can add features that use the OctoBot's object detecting sensors, buzzer, and even the self-charging routine.

See the OctoBot manual for details and — of course — use this information as a starting point for creating your own circuits and programs to add on to the OctoBot.

Summary

The Internet Age brings with it

Figure 8. Final position of the line tracker on the bottom of the robot. Note that the three sensors are positioned toward the front of the robot.

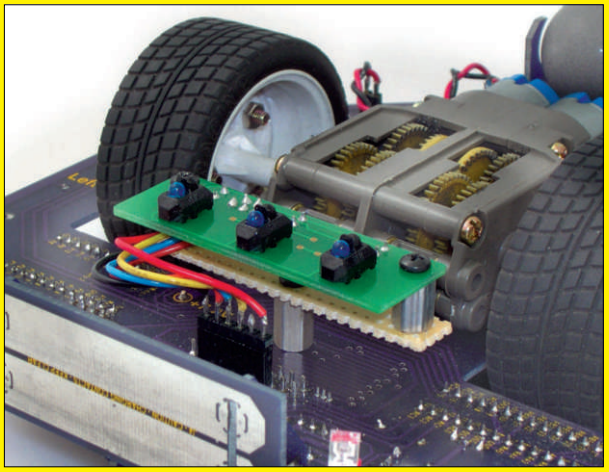
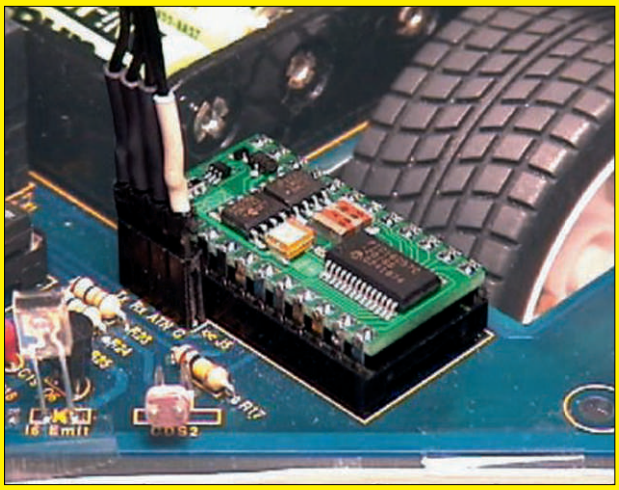


Figure 9. The Stamp 2 and serial cable on the top side of the robot.



many amazing extensions to our abilities to research, learn, and interact with others. With it, we can have many friends and business associates whom we may seldom or even never actually meet in person. However, nothing replaces the benefits of face-to-face human contact. For robot builders, belonging to a local club (or starting one, if you can't find one in your area) can provide important information, as well as many good times spent with like-minded explorers.

The two clubs that I first joined long ago continue to meet and grow, both in the real world and in cyberspace. Join them both from wherever you get Email:

• HomeBrew Robotics Club, San Jose, CA — **www.hbrobotics.org/**

• Robotics Society of America, San Francisco, CA (currently helmed by regular *SERVO* columnist David Calkins) — **http://groups.yahoo.com/group/sfrsa/**

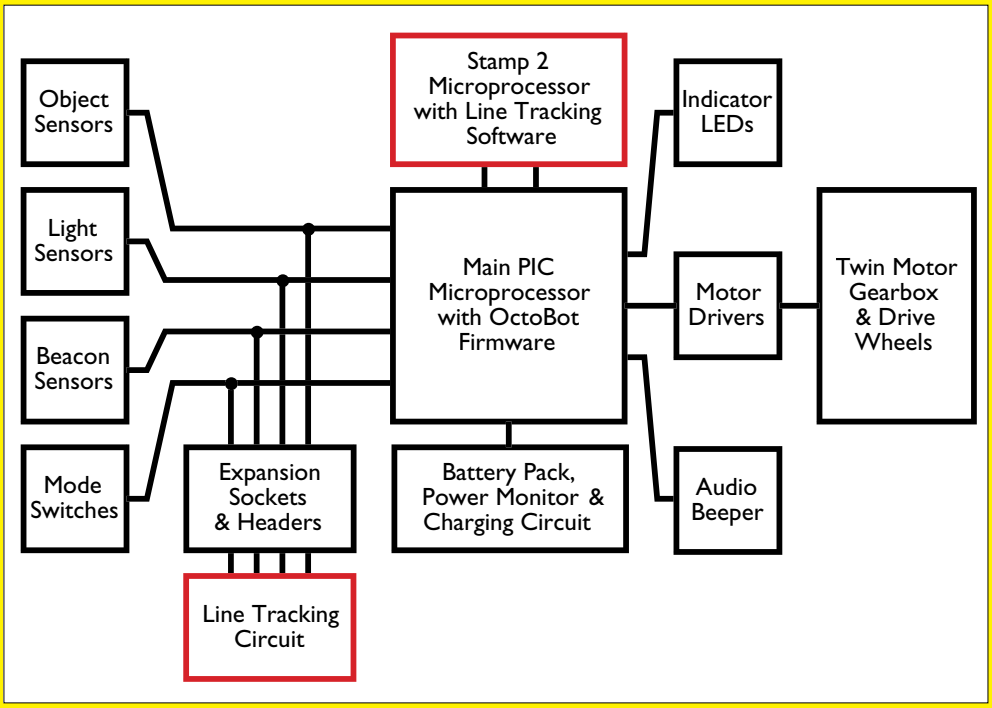


Figure 10. Block diagram of the OctoBot with Stamp 2 and line tracking add-ons.

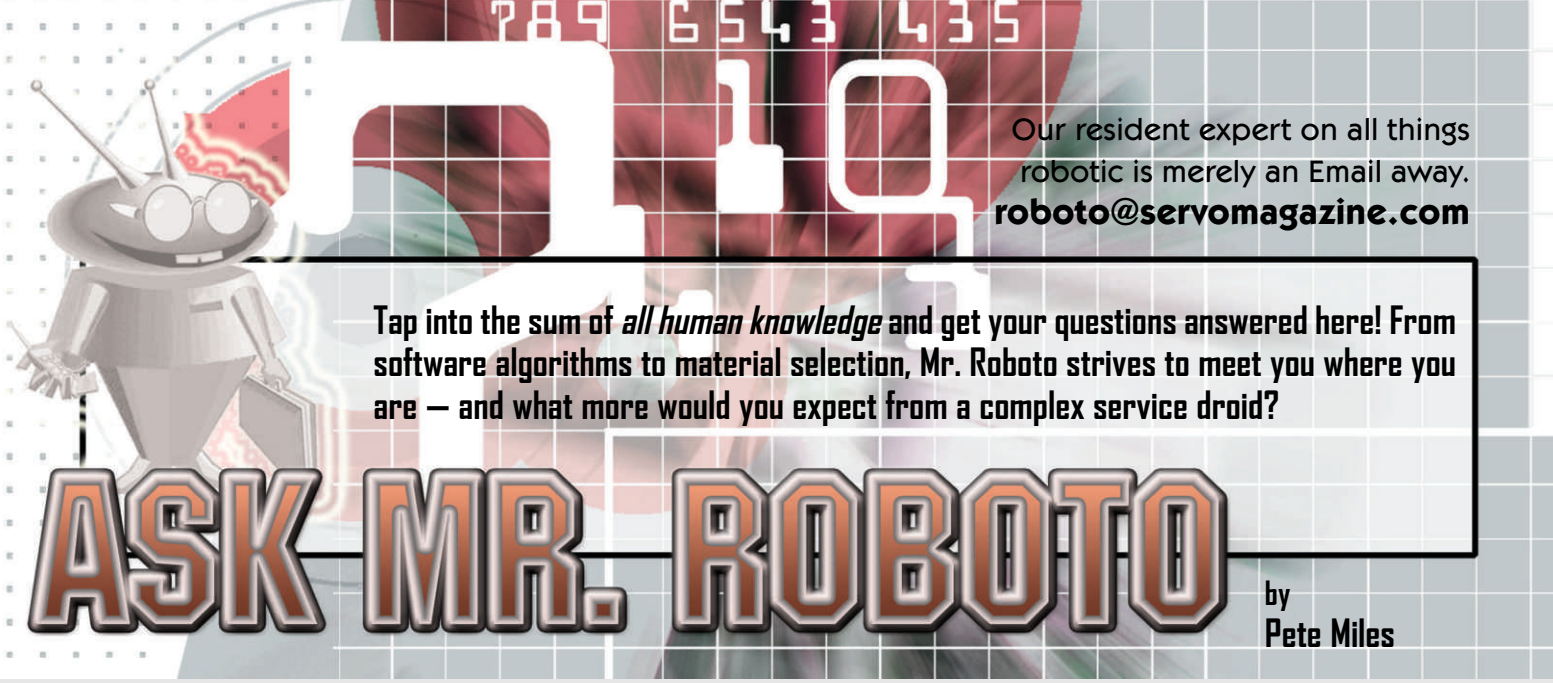
(If you're looking for a local robot club to join, visit the SERVO website and click on the "Robot Clubs" link on the left. — *Editor Dan*)

I still have regular contact with many people I've met over the years through these clubs. Although the online world can provide access to robotics content around the globe, nothing replaces the benefits and pleasures of real world interactions with other robot-building humans. So, find a club, join in, and ... build more robots! **SV**

About the Author

In college, Roger G. Gilbertson studied engineering, robotics, and the walking patterns of living creatures. In 1987, he co-founded Mondo-tronics, Inc., to explore the commercialapplications of Shape Memory Alloy wires and, in 1995, he launched RobotStore.com, the Internet's first commercial robotics site. Mondo-tronics' Robot Store continues to lead the field in presenting the best and most innovative new robot products for students, educators, hobbyists, and experimenters. Roger lives and works in Marin County, CA, where — unfortunately — robots have yet to take initiative to help us humans keep our world in good shape.

HOBBY ENGINEERING
The technology builder's source for kits, components, supplies, tools, books and education.
Robot Kits For All Skill Levels
ICs, Transistors, Project Kits
Motors, Frame Components and Scratch Builder Supplies.
Books and Educational K
BEAM Kits and Components
Order by Internet, phone, fax or mail.
www.HobbyEngineering.com
1-866-ROBOT-50
1-866-762-6850
1-650-875-0715
1-650-952-7629 (fax)
sales@HobbyEngineering.com
1405 Huntington Avenue, Suite 150
South San Francisco, CA 94080
Visit our showroom near SFO!
Most orders ship the day received! World-wide shipping. Convenient payment options.



Our resident expert on all things robotic is merely an Email away.
roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

ASK MR. ROBOTO

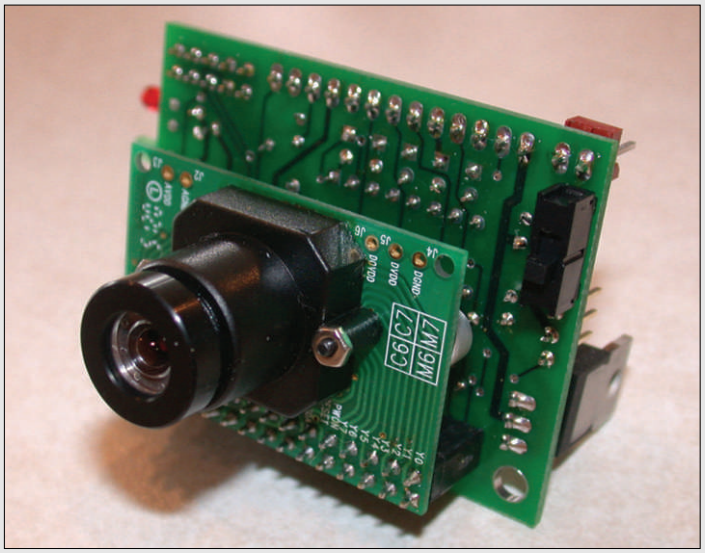
by
Pete Miles

Q. What is the difference between a servo and a gear motor?

— Mike Hamming
Schenectady, NY

A. The one thing that servos and gear motors have in common is that servos have gear motors built inside of them. A gear motor consists of an electric motor and a set of gears to reduce the speeds of the motor's output shaft and increase the overall torque of the motor. A servo also has a motor and a set of gears, just like a gear motor. The main difference is that a servo motor incorporates a feedback and control system. The system monitors the output shaft angular position or velocity (or both), compares the actual results to the desired results, and then the control circuit adjusts the servo motor until the results match. So, gear motors do not incorporate feedback devices and servos are gear motors that incorporate feedback devices.

Figure 1. The original CMUcam.



Q. I'm thinking of implementing some kind of machine vision in my robot by means of an eight-bit microcontroller and a camera. The problem is, I haven't found any suitable cameras. The ones I have found either have composite analog output or are overly complex to interface to a microcontroller.

Since eight-bit micros don't have many resources, I need one that is easy to interface and has a low resolution. Can you help me?

— Tobias Reneskog
via Internet

A. Take a look at the CMUcam developed at Carnegie Mellon University (**www.cs.cmu.edu/~cmucam**). The cameras have been licensed to Acroname (**www.acroname.com**) and Seattle Robotics (**www.seattlerobotics.com**). These cameras are ideal for using with eight-bit microcontrollers. All of the data that is transmitted to and from the camera is through standard three-wire RS232 serial communication using the eight data bit, one stop bit, no parity protocol. Table 1 shows a short comparison between the two camera models. The original CMUcam has been around for several years now and the new CMUcam2 has just been released. These cameras have become one of the easiest and more popular methods of implementing vision into different robotic projects.

One of the more popular features that these cameras have is the ability to track objects. The camera can be set up to track the first color object it sees or it can be programmed to track a specific color or the brightest object in the field of view. The camera can return the coordinates of the centroid of the color object that is being tracked or it can control R/C servos that can be mounted to hold the camera so that the camera will follow the colored object of interest. This makes the camera ideal for Fire Fighting and Soccer playing robots.

The CMUcam2 has the advanced ability to compare two images (Frame Differencing) to see if an object has moved between the two images and then can use this information to track that movement.

The CMU website has many sample programs for using this camera, including a Java application that allows your computer to send and receive data to and from the camera and to be able to see the exact images the camera sees. In order to use these applications, you need to have Sun's Java language installed on your computer. (If you don't have Java, you can download it for free from their website, **www.java.com**) When using Java, remember that everything you type is case sensitive. For example, typing `cmucamgui` will not run the `CMUcamGUI` application program that comes with the camera because it does not match the exact typed case.

I am pretty sure that you will be able to do most — if not all — of the things you want to use the camera for. Just keep in mind that implementing vision in a robot usually turns out to be more complicated than most people originally anticipate. Once you have it up and running, however, you will be amazed at all of the new capabilities your robot will have.

Q. Hi. I have been a subscriber of the fantastic *SERVO Magazine* since February and enjoy your educational and informative articles, but let me get to the point. I have converted at least 10 manual milling machines to operate by CNC using stepper motors without encoders. Now, I want to build a servo-powered machine and need a book about encoders that explains how they work, how to test them, and how to use them.

I looked at your library, but could not find encoders listed. So, my question is, do any of your books have encoder information in them or could you point me in the right direction?

— Don Bolton
via Internet

A. The best book from the *SERVO* Bookstore for explaining how encoders work and how to use them is the book *Building Robot Drive Trains*, by Dennis Clark and Michael Owings. About 60 pages of this book are devoted to this topic. In my opinion, it is one of the best single-source explanations of this topic out of all of the robotics books I have seen. Though the closed loop PID (Proportional-Integral-Derivative) algorithms are oriented for velocity control, the same methodology can be applied to position control, which is just as important — if not more important — than velocity control in CNC (Computer Numerical Control) applications.

I would also recommend that you read the various application notes located at National Instruments (**www.ni.com**). Use a keyword "Encoder" in their search engine. There are a lot of little, subtle things that are important for CNC applications that are not normally dealt with in the mobile robotics community. They affect the long term repeatability and accuracy of the overall machine, along with handling acceleration and deceleration issues with axis direction and velocity changes and circular interpolation. Though the information is oriented toward using the data acquisition and motion control boards and LabView software, the various application notes do explain quite a few of the issues that are involved with machine tool motion control.

I would also recommend that you join the CAD_CAM_EDM_DRO Email group at **www.groups.yahoo.com** There are over 5,000 members who are doing the same type of work that you are doing. On that Email group, you can post questions regarding where to learn about using encoders and they will provide a lot of good information. They are also very helpful in providing information that will be helpful in developing the software that you will need to use with the encoders and your CNC machines.

Q. My robot uses the GP2D12 analog sensors and I am having a hard time trying to filter the noise from them. Do you have any suggestions?

— Mike Owens
Atlanta, GA

A. Many people in the robotics community reduce the noise problem in the Sharp GP2D12 sensors by grounding the plastic case on the sensor. It is an odd technique, but it does seem to work. In my experiments, I

Feature	Original CMUcam	CMUcam2
Tracking frame rate, frames/sec	17	50
Baud rate	9,600-115,200	1,200-115,200
Number of servos that can be controlled	1	5
Camera resolution	80 x 143	176 x 255
Built-in microcontroller by Scenix	SX28	SX52
Analog video out	✗	✓
Find centroid of a tracking object	✓	✓
Obtain color statistics of the image	✓	✓
Power down mode	✗	✓
Dump raw image	✓	✓
Remotely adjust camera imaging properties	✓	✓
Number of automatic target tracking axes	1	2
Omnivision CMOS camera	OV6620	OV7620
Overall size	2.25 x 1.75 x 2	2.5 x 2.2 x 2.2
Power	6 V, 200 ma	6 V, 200 ma
List price	$109.00	$219.00

Table 1. CMU camera selected feature comparison.

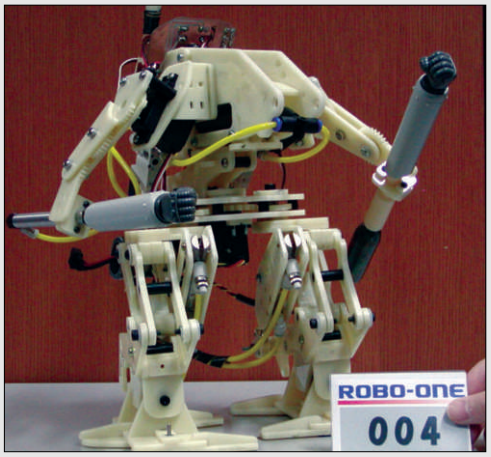


Figure 2a. Robo-One style robot.

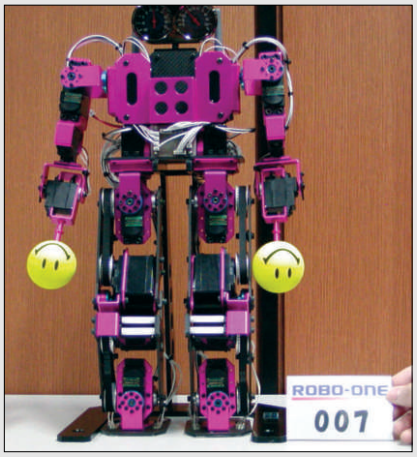


Figure 2b. Robo-One style robot.

have seen peak-to-peak voltage variations of 1.2 volts when no object is in front of the sensor and the sensor case was not grounded and — when I grounded the sensor — the peak-to-peak variations dropped to about 300 mV. With an object that is 12 inches away from the sensor, I have measured voltage fluctuations of 600 mV without grounding the case and variations of 150 mV with the case grounded. It is hard to believe that grounding a plastic case can do all this, but it does.

The plastic seems to be made from some sort of a conductive material. I used a multimeter to measure the resistance across the sensor case at the screw mounting terminals and the results I obtained showed a resistance that ranged from 200 to 600 ohms with the different sensors that I have. On one of my sensors, I applied 5 volts across the screw terminals on the sensor's case and measured 37 ma of current going through the case — very surprising.

Most likely, the noise is a by-product of the 200 ma instantaneous current draw the sensors have when the LED fires that is stressing the rest of your power circuit. However, this case grounding technique does reduce the voltage fluctuations in the analog signal line and is widely used throughout the robotics community. Grounding the case is a quick and easy method that will help with simplifying the software in your microcontroller to handle false signal changes due to the fluctuations, since the fluctuations seem to be cut by a factor of 4.

Q. I heard that there is a new boxing robot contest in Japan. Can you tell me more about it?

— Joseph Williams
Calgary, Alberta

A. You must be talking about the Robo-One contest (**www.robo-one.com**). This is probably one of the coolest robot contests ever created. Here, two biped robots compete in hand-to-hand combat that includes pushing, boxing, and karate — with the goal of knocking the other robot over.

These bipeds are extremely versatile. They generally have over 20 servos in them and they walk just like humans do. They can balance on one leg, walk, run, and hop. They can do push-ups and hand stands. They even can get back up on their feet when they fall over.

The Robo-One website is in Japanese text, but you can find many pictures of the different robots that have competed in this contest since it started in 2002. There are many links to the home pages of the robot builders and there are quite a few video clips showing these robots in action. This site is one of those prime examples that a picture is worth a thousand words. You don't need to understand Japanese to be amazed at what these robotic creations can do.

The rules for the boxing (wrestling) contest are relatively simple. Each match consists of three rounds. Each round is two minutes long. When the robot is knocked down, it is given a 10 count. If the robot cannot get back up on its feet within the 10 count, it is a knock out, the round is over, and its opponent is awarded the victory for the round. If the robot in knocked down three times in a single round, it is considered a knockout, and the robot's opponent is awarded the victory. The robots are given two minutes between each round to make any repairs. The robots must make two complete steps before they can attack; this includes getting up after being knocked down. The robots must not remain stationary for more that three seconds. The competition ring is a four foot square that is painted white.

There are currently no weight limitations to the robots. Their height must be between 20 (8 in) and 120 cm (47 in). The maximum length of a foot is 20 cm (8 in), but it also cannot exceed the length of a leg segment. When standing, the perimeter of the feet cannot overlap each other. The robot can be either autonomous or remote controlled. Control can also be through wireless communication between the robot and a computer. Robots cannot electrically (includes radio) or optically interfere with the control of the other robot. Robots cannot have weapons — such as knives or spinning devices — that can injure humans. The robots cannot have any device that can damage, mark, or deface the competition ring. The robots cannot have any suction devices on their feet, cannot blow liquids, powders, or gasses at their robot competitors, and they are not allowed to have any form of a firing device.

The current official rules for this event are located on the official Robo-One website. An English translation of the rules is located on the Robolympics (**www.robolympics.net**) website. In March of 2004, the Robolympics hosted the first Robo-One event in the US.

These are some of the coolest and most exciting robots I have ever seen. Their only drawback is that they are not easy or inexpensive to build. They require the high strength,

high performance servos and servo controllers that can handle over 20 servos at the same time, along with a lot of CPU power and memory. There are a few companies in Japan that sell Robo-One, including Mugen-Kobe Co., LTD (**www.ne.jp/asahi/robo/fantom/**) for about $2,000.00 to $4,000.00 each. In the USA, SOZBOTS has just started carrying the Kondo KHR-1 at **www.robo-one-usa.com**

The Open Pino project (**www.openpino.org**) is an open source project with complete plans for making one of these robots. You have to register with them to get a password to log onto their site. The last time I checked this site, it was asking for the password before you can register to get the password. Since it is the same for everyone, the ID is pino and the password is open-pino. When you download the drawings, you will have a good idea on how these robots work on the inside. It's not that complicated — just a lot of sheet metal work.

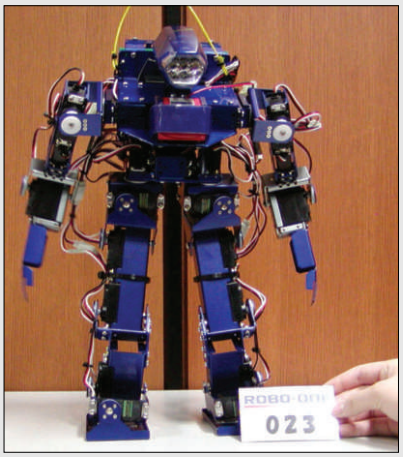


Figure 2c. Robo-One style robot.

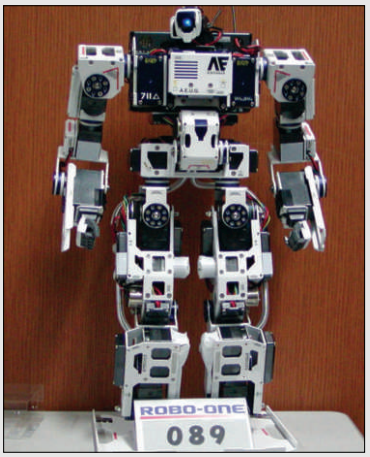


Figure 2d. Robo-One style robot.

Q. I am in the process of looking for a college to go to after I graduate next year. I really like walking robots, so I am planning of getting either a mechanical or electrical engineering degree. Are there any colleges that you can recommend that have walking robotics programs?

— Steve Phillips
via Internet

A. I generally recommend that people take a serious look at their local in-state colleges and universities before looking at out-of-state schools. This is mainly because of the cost differences between in-state and out-of-state tuition. When you are in college, the quality of the education that you get solely depends on you and how much you want to learn; it does not depend on the school you go to. Make sure that the engineering program you are looking at is ABET accredited.

As for walking robot programs, well, that depends on what research projects are going on at the time you plan to go to the school. Make sure you ask for a list of research projects the faculty is currently working on and then contact the professors who are working in this area to find out if they anticipate continuing to work on the projects over the next five years. A lot of research grants are only a couple of years long, so today's research may be completely different from what is happening in three years.

Now, with that said, there is another way to work with walking robots in college. There is an international robotics contest sponsored by the Society of Automotive Engineers called the Walking Robot Challenge (**www.sae.org/students/walking.htm**). The 2004 walking robot challenge consisted of seven events. The first event — which every robot team must participate in — is the Design Event, which has four parts: Technical Paper, Technical Presentation, Design Evaluation, and Poster Paper. The other six events include: the Dash, Load Retrieval, Slalom Course, Trip Wire, Object Seeking, and the Endurance and Obstacle Challenge.

This is a very exciting and challenging event and all engineering programs should consider participating. The 2004 event had 19 different schools involved. You can look at the past events to see which schools routinely enter the event. If the school you want to attend has not participated in this program before, you can form a team to enter the contest, since it is open to all schools. SV

TETSUJIN TECH COMPETITOR PROFILE

Here is a sneak peek at the teams currently building for *SERVO Magazine's* powered exoskeletal weight lifting competition. For more information on the event, visit **www.servomagazine.com/tetsujin2004/**

The RoboLancers

Philadelphia, PA

Team Leader: Zach Powers, student

Background: We are a high school team comprised mainly of students, a few adult mentors, and a few team alumni who are now in college. We are all highly dedicated, love to work with robotics, and love to build and innovate. Our main focus is building for FIRST every year. We struggle to find the resources — monetary and otherwise — to enter into several competitions, but we always make a great showing. As the head of the team, I would be hard pressed to find another group of people who even approach the level of dedication, intelligence, ingenuity, and tenacity that this team exhibits.

Motivation: We want to expand our horizons in the types of machines we build and, of course, a few of us have just been waiting for a reason to build an exoskeleton!

Strategy: We may not lift the most weight, but we plan to lift it quickly, reliably, and safely. We will manufacture an exosuit that can repeatedly lift the needed weight quickly, while still allowing the user freedom of motion.

Largest Obstacle: The biggest obstacle for our team will be gathering money to fund our efforts. We are a high school team in Philadelphia and it is hard to find the money we need, as well as the space we need to build in.

Academic Focus: My plans are to attend either Drexel University or the University of Pennsylvania.

Construction Materials: We plan to build our suit from aluminum and carbon fiber.

Power Source: We are going to power it with nickel metal hydride batteries.

Estimated Cost: $3,000.00

Contact: **www.centralhigh.net/~robot/robotics.html** or through Email at zmaster265@hotmail.com

Binary Bot Boyz

Hurst, TX

Team Leader: Billy Holcombe, machine electronics technician, Bell Helicopter Textron
Team Engineer: Fred Ritz, control systems engineer

Background: I am 56 years old and built my first computer in 1975. I enjoy amateur radio, electronic gadgets, scuba diving, and off-road motorcycling. My job at Bell Helicopter involves installation and service on state-of-the-art control systems for manufacturing parts for the V-22 Osprey. I previously worked on Flexible Manufacturing Systems for the F-16 program with Lockheed Martin.

Motivation: I am entering Tetsujin 2004 because I love a challenge.

Strategy: I think a KISS (Keep it simple, stupid!) centered strategy will be best.

Largest Obstacle: My biggest obstacle will be fitting everything into a reasonably sized package and to engineer a system that allows for balance under dynamic conditions — the lift and subsequent lowering of the load.

Academic Focus: I attended Tarrant County College and studied electronics and programming.

Construction Materials: Primary construction materials will be carbon fiber composite with honeycomb reinforcement, aerospace fasteners, and unitized construction techniques.

Power Source: I will be using DC electric motors powered by gel cell batteries, which have an excellent cost/performance ratio.

Estimated Cost: unknown

Contact: bfholcombe@charter.net

This Is One Event You Don't Want to Miss!

Tetsujin 2004 is being held during the RoboNexus International Conference and Exhibition, October 21-23, 2004. For registration and attendance information, visit **www.robonexus.com**

DARPA Grand Challenge Spawns SRS/SERVO Magazine Robo-Magellan—Part 2

by Michael Miller

In this second part of this series on my robot entry for the SRS/*SERVO Magazine* Robo-Magellan contest, I will discuss the four key detection systems that I used on my bot: obstacle, direction, location, and target detection. In last month's article, I stated that my overall solution is to rely on a set of pre-prepared waypoints that my bot must navigate to and to ignore direct GPS readings, except when an expected target cone is not found. I believe this approach will allow me to keep the systems as simple as possible while using only off-the-shelf components.

Obstacle Detection

For obstacle detection, I decided to use a combination of methods that overlap to provide a resilient and simple detection system. Ranged detection will be handled by Sharp IR range sensors, while a set of whiskers will handle anything that gets past the IR system.

On the front of the robot, I need to be able to detect not only walls or obstacles, but also falloffs. The use of the range sensors allows me to get a distance from an object and the ground. By facing the sensor down in front of the bot, the range of the ground can be obtained. If the range gradually increases or decreases, this means the robot is approaching a slope in the ground. A sudden increase means a hole, stair step, or a curb and a sudden decrease means it is approaching an obstacle (see Figure 1). The exact range isn't interesting; it is the relative reading of one of the three states — Drop, Flat, or Rise — that is important. Also, the distance between the target and the front of the robot must be within the middle of the range of sensors so that the target is far enough in front of the bot that it can react to the sensor input changes.

Before determining where to place the sensors, I needed to know how much distance the bot required to come to a controlled stop from its top speed because this represented the worst case scenario. For control purposes, I decided to limit the top running speed to 1 meter per second. My tests showed that it requires no more

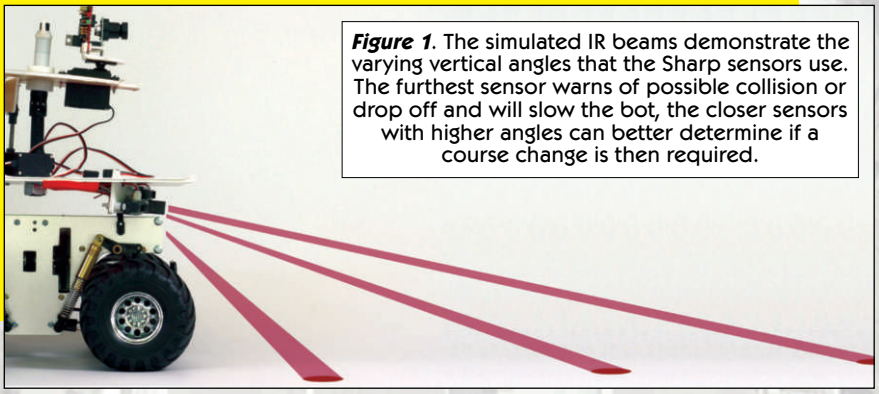


Figure 1. The simulated IR beams demonstrate the varying vertical angles that the Sharp sensors use. The furthest sensor warns of possible collision or drop off and will slow the bot, the closer sensors with higher angles can better determine if a course change is then required.

than 30 cm to stop on loose material when initially moving at 1 m/s. I found it very handy to use a remote control transmitter and receiver to conduct these tests.

Next, I needed to know how much time the sensor itself took to sample a reading and make it available to the microcontroller. The data sheets for the analog Sharp IR sensors state that they have a pretty wide range of 38.3 ms ±9.6 ms to take a reading and a maximum 5 ms latch time to update the voltage output. Thus, the worst case time to take a distance measurement would be 53 ms. When the robot is traveling at maximum speed, it would travel about 5.3 cm by the time the sensor outputs a new measurement result.

Although my analog-to-digital converter took 10 ns to convert the analog signal into a digital signal, I decided that I only needed to update the sensor data at 25 Hz (once every 40 ms). Thus, when my robot was moving at maximum speed, the bot would only travel 4 cm between each sensor update cycle.

By adding all these delay times (stopping distance, time to take a sensor reading, and sensor update time), I can determine the distance my sensors need to be monitoring to ensure that my robot can react to its environment. Based on my calculations, the sensors need to be able to detect a rise or fall of 39.3 cm in front of the bot when traveling at top speed.

My forward sensors were mounted 15 cm above the ground. Using the Pythagorean Theorem ($a^2 + b^2 = c^2$), I calculated the range from the sensor to flat ground that would be needed to be about 36.3 cm. At this range, all the Sharp IR sensors are possible candidates. The GP2D120 has a range of 4-30 cm, the GP2D12 has a range of 10-80 cm, and the GP2Y0A02YK has a range of 20-160 cm. The 36.3 cm middle of the sensor detection range goal does not fit the GP2Y0A02YK, as it is barely above the 20 cm minimum. Further, since the IR sensors have a very small detection area (the area the IR beam would be reading is less than the size of a nickel, even at a distance), the bot would need several of them to give a better view of the forward obstacles, especially while turning (Figure 2).

My solution was to use five sensors instead of just one. One longer-range sensor in the middle is aimed further out for when the robot is at higher speeds while traveling straight. I have two shorter-ranged sensors — each aimed to cover the closer and side areas directly in the path of the tires. Then, I have two very short-range sensors aimed even further to the sides that will be useful for sharp turning (Figure 3). I opted for one GP2Y0A02YK aimed at about 50 cm out with a flat ground range of 52 cm, which is well into the middle of its range. Also, I added two GP2D12s aimed about 25 cm out, with a flat ground range of 29 cm. Then, I added two GP2D120s aimed about 15 cm out, with a flat ground range of 21 cm.

The center sensor, of course, would be aimed straight along the central axis of the bot and mounted to point forward. I wanted to use the closest two sensors while turning sharply, so I used the turning angle of 17° and mounted them above the tire turning axis. The other two mid-ranged sensors are targeted along the tracks of the tires when the bot is traveling straight forward.

The way I elected to mount them was to build a rail that all the sensors would sit on. This rail would be mounted under the front lip of the base. It would be an arc of a circle of at least 34° (double the turn angle) from center of the left and right short range sensors. I then needed to calculate the radius of this circle so that I could find the arc length of this rail and build a jig that will help me construct it.

By using the locations of two short range sensors and taking into account that they sit on this rail 34° from each other, you can calculate the unknown center point in a right triangle using $\sin \alpha$ = opp/hyp — in thie case, sin(17) = 7 / r. The radius was therefore 23.9 cm. The arc length of the rail is calculated by $l = r \cdot |\alpha|$ (absolute value of angle in radians), which — in this case — comes out to be approximately 14.2 cm. This provides the distance along the rail from the centers of the two outermost sensors. Add the width of the sensors (4.5 cm — half to each side) to get 18.7 cm and then add a little for structure past the sensor mounts to get 20 cm for the rail length. This design was then formed onto a cylinder with the radius of 23.9 cm to create the sensor mounting rail.

The locations of the sensors on this rail are easily defined

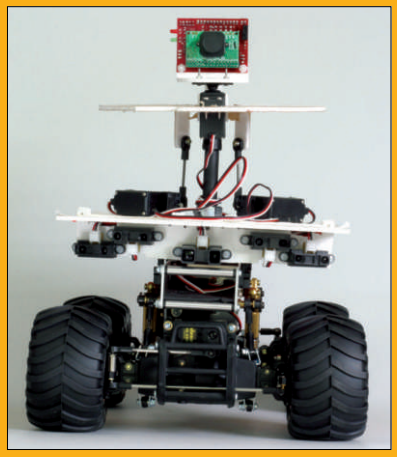
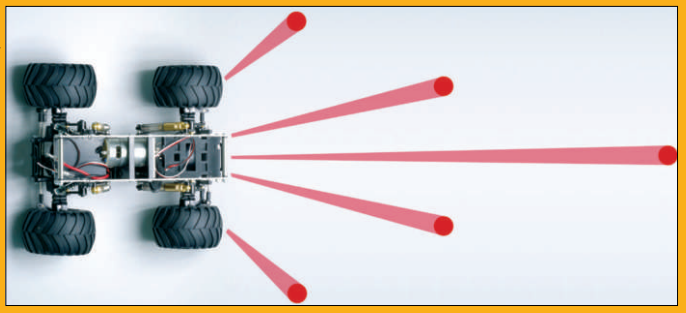


Figure 2. This front view of the bot shows the five Sharp sensors mounted under the primary platform for protection. The arced mount provides the forward directional angles for each sensor.

Figure 3. The simulated IR beams demonstrate the horizontal spread of the Sharp sensors. A fairly complete coverage is achieved with five sensors. The outer two provide coverage in sharp, slow turns, the center provides warning while moving quickly, while the middle two cover the path of the tires.



for the center sensor (at the center) and outermost two (at 14.2 cm along the arc). The midrange sensors require a little more calculation, as they will be mounted on this arc and I wanted them aimed along the tire tracks about 25 cm in front of the bot. I needed to find the arc angle and, for this, I need the lengths of two sides of a right triangle. The first side is half of the tire track width (10 cm).

The other is the distance that the arc circle center is set back along the bot to the sensor target. This length is made up of the 25 cm forward distance and the radius of our arc circle (23. 9 cm), which comes to 48.9 cm. I calculated the hypotenuse of the triangle to be a length of 49.9 cm and, again using sin α = opp/hyp to find the measure of the opposite angle, I calculated the arc angle of the sensor to be 11.5°. This gave me an arc length from center of about 4.8 cm and the location on the rail to mount the sensor.

Now that the sensor locations on the rail have been determined, the sensors themselves need to be angled downward. This is accomplished by using wedge washers that have the angles required for either side of the sensor to mount it to the rail. I could have calculated this angle with the information I have and I did this for the starting point, but the errors in my craftsmanship that occurred when making the sensor rail made the values useless.

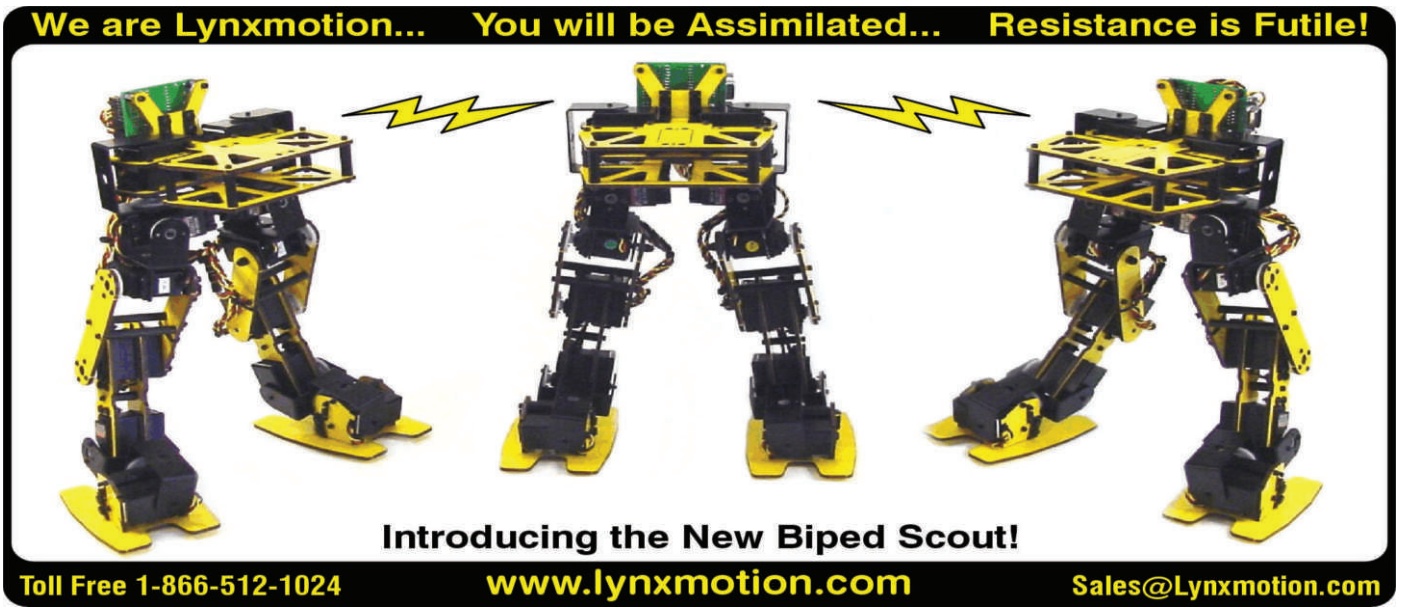
After assembly, I needed to set the sensor target due to all of the mounting errors. I used an old black and white bullet digital video camera to view the infrared sensor target locations. Most low cost video cameras are sensitive to infrared light and the IR will show up as a well-lit spot — just as though you were pointing a flashlight at the ground. With this information, I was able to adjust my mounts and angles to get the final correct locations.

The actual readings from these sensors needed to be calibrated into the three states that I am interested in: Drop, Flat, and Rise. Drop will represent a significant "drop" in the terrain beyond what I want the bot to attempt to go down. Flat will represent the normal operating level, although being completely flat is a relative term. A Rise will represent a wall or steep uphill ramp beyond what I want the bot to attempt to climb.

My method to calibrate these states is to use a sheet of wood that I place directly in front of the bot and angle it up and down while sampling sensor data for each set of sensors. Since most of grounds of the Seattle Center are wheelchair accessible, I used the standard wheelchair ramp specs as a guide to what I should set these levels to. A standard wheel chair ramp is a 1" drop for every foot — about 4.8°. I felt that some error would be appropriate on top of this, so I doubled it to about 10° and I feel that I should be able to find paths within this range and the bot should be able to handle them.

One limitation of the sensors used in my design is that they are not very good at sensing very thin or inconsistent surfaces, as they are only sampling the terrain in five nickel sized areas. Like the DARPA Grand Challenge entrants that needed to sense barbed wire, this bot must be able to handle tall grass or plant stalks that — if they were to go undetected — could stop the robot dead in its tracks. Although it seems simple, adding a set of whiskers to the front of the bot will help detect anything that unintentionally or — in the case of the target traffic cones — intentionally gets through the range sensors. Any obstacles that fall between the range sensors will either be sensed with the whiskers or — in the case of small rocks or clothing — will be handled by the suspension of the bot and simply be driven over.

To build these whiskers, I used a standard approach of attaching a stiff, but lightweight, wire to the lever of a micro switch and mounting them on the front axle of the bot, above the tires. By mounting them on the axle, they stay in line with the tires as they articulate on the ground. Mounting them above the center line sets the height of change to a level that the tires and suspension will not be able to handle.



I cross-mounted the two whiskers allowing directional detection of either side with complete coverage across the front.

Direction Detection

As part of my approach, it was important to know which direction the bot was facing at all times. The best way to do this is to add a compass. I looked at several microcontroller-friendly compass solutions and decided on the Dinsmore R1655 analog compass sensor. It is relatively inexpensive and provides the magnetic direction as two analog signals that can be easily converted into a degree heading. I also chose the 1655 over the 1555 due to wider analog output of ±0.75 V around the 2.5 V center that will provide higher accuracy when using the ADC. Another nice feature of these sensors is that they draw only 19 ma when active and only have a rise time of 90 ns, so they can be managed as needed to use very little power.

The sensor comes in a relatively small cylindrical package with a 12.7 mm diameter and 16 mm height. The leads protrude out of the bottom (Figure 4). It needs to be mounted as far from any bot-generated magnetic interference as possible, including not only motors, but also any large ferrous metal pieces. One very nice feature that it does have is internal dampening on the readings, so any short but erratic readings are not an issue. However, it does require that it be held within 12° of level to generate an accurate angle. With the expectation that the bot should be able to handle slopes greater than this, a solution would be needed to maintain a level base for the compass.

With the idea that other sensors may also want a more level mounting location, I decided to create a head that would actively level itself (Figure 5). It needed to be able to tilt in any direction and clear the base. At my local hobby shop, I found a set of U-joints with drive shafts mounted as replacement parts for a hobby R/C monster truck that would provide the joint and mast. I threaded the ends of the drive shafts so that it could be mounted to the base and head platform. I made the head platform from a plastic sheet that I cut to template; I formed tie rod connection tabs with a hot air gun. This U-joint bore the weight of the head platform and kept it from rotating.

The active leveling was provided by two servos mounted with tie rod ends. The servos are mounted to the base unit and the ends of the tie rods are mounted to the tabs I formed on the head. These servos are controlled by the microprocessor to keep the head level, based upon readings from a tilt sensor.

I chose the ADXL202 accelerometer as a tilt sensor. It is sensitive enough to pick up on the gravity of the Earth and provide an accurate tilt angle while providing a fast response to keep the head within the 12° error allowed while in motion. I used the evaluation board (ADXL202EB) so that I did not need to create my own circuit to interface to it. I used the PWM output from the sensor so that I didn't need to tie up another ADC on my microcontroller. Although the documentation states that the frequency will wander as the temperature changes, my method of reading PWM provided both the time of the pulse width and the trailing non-pulse width, so I relied on the relative reading of the pulses to the frequency, since the exact angle value wasn't interesting because all I needed to do was to keep it centered.

Location Detection

Although my bot will start with the path to the target cones as a series of heading and distance measurements, it needs to know when it reaches each waypoint in this path so that it can change its course to the next leg of the path. Actively reading a GPS for location could be done, but the error is much higher than what could be achieved with dead reckoning methods. The compass will provide a fairly accurate heading, so all that will be needed is a method to calculate distance traveled.

Calculating the distance traveled is not always as easy as it sounds. Many bots will have an encoder on the motor to accurately measure the speed and, thus, provide distance, but this is prone to errors due to wheel traction issues. Further complicating this is that my bot's chassis has differentials.

A differential (in terminology of vehicles) is a gear system that transfers powers from an input shaft to two output shafts

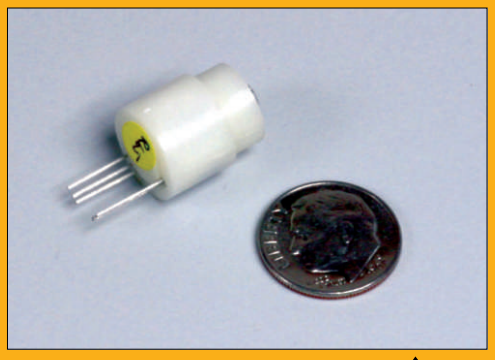


Figure 4. The Dinsmore R1655 analog compass module is both small and accurate.

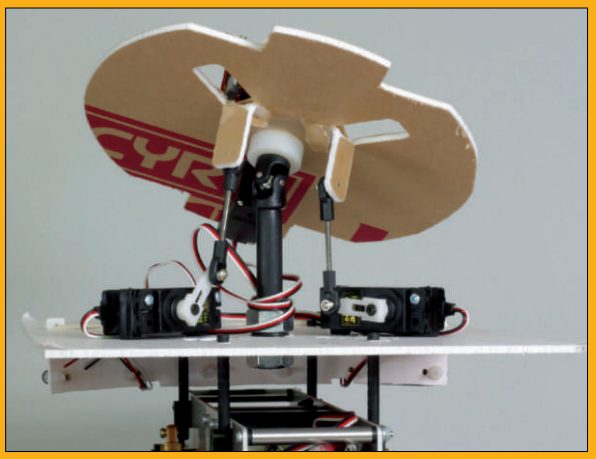


Figure 5. The auto leveling head uses two servos mounted 90° against each other. They will "pitch and roll" the head platform to keep it level.

Decoding the GPS Data

There are many data sentences that the Garmin GPS 18 will send, but I wanted to demonstrate the few that I used to handle all my needs. The sentences all use visual ASCII to represent values. The GPS unit follows the NMEA -0183 format, which uses 4800 baud, eight data bits, one stop bit, and no parity. See the online document at **www.garmin.com/manuals/GPS18_TechnicalSpecification.pdf** for the full specification.

All sentences sent start with a dollar sign ($) and end with a line feed (10) and carriage return (13) pair. Although it is sometimes optional, most sentences will have a checksum that will precede the line feed. The checksum starts after the asterisk (*). The GPS unit will continually send updated data once per second.

GPGGA – Global Positioning System fix data received:

```
$GPGGA,160501,4807.038,N,01131.324,E,1,08,0.9,54
5.4,M,46.9,M,,*42
```

160501	Fix taken at 16:05:01 UTC
4807.038,N	Latitude 48 deg 07.038' N
01131.324,E	Longitude 11 deg 31.324' E
1	Fix quality: 0 = invalid, 1 = GPS fix, 2 = DGPS fix
08	Number of satellites being tracked
0.9	Horizontal dilution of position
545.4,M	Altitude, in meters, above mean sea level
46.9,M	Height of geoid (mean sea level) above WGS84 ellipsoid
(empty field)	Time in seconds since last DGPS update
(empty field)	DGPS station ID number

PGRME – Garmin proprietary sentence, error received:

```
$PGRME,15.0,M,45.0,M,25.0,M*22
```

15.0,M	Estimated horizontal position error in meters (HPE)
45.0,M	Estimated vertical error (VPE) in meters
25.0,M	Overall spherical equivalent position error

(often axles) and allows each of the output shafts to turn at different speeds, based upon resistance to turning. This is a great feature for vehicles. When making a turn, the inner wheels will turn less than the outer wheels and keep the wheels from slipping on the terrain or causing undue wear on the tires.

Unfortunately, not only can taking turns cause the differentials to slip, but the wheels can slip, also. This can happen when one side is on wet grass and the other is on concrete. This requires that the encoder be placed on the output side of the differential – on the wheel. Since each wheel can turn differently relative to the anothers, encoders are required on each one.

Since the wheels are going to be the part of the bot that is exposed the most to the environment, I felt the encoder sensors needed to be able to handle a little dirt and water. I chose to use magnetic sensors as a result. I embedded small magnets inside the wheels and used the sensors mounted to axles to read the magnetic field as it moved past. This is very similar to most modern automotive anti-lock brake sensors.

I chose a very small, but powerful, button-shaped magnet. These easily slip into a small indentation I drilled on the interior side of the wheel, which was then locked in place with super glue. Mounting the magnets on the interior of the wheel protects them from the environment. I placed them evenly around the wheel at 72° from each other so that I had five readings per turn.

The sensor I used to detect the magnets was the AD221-00 from NVE Corp. It is a small surface mount chip that – based upon the presence or absence of a magnetic field – will switch the output pins. This specific chip can source or sink a signal, so interfacing it to a microcontroller was a direct connection. I built a small circuit board that mounts the chip to the axle so that it can read the magnets embedded in the wheel. I dipped the sensor board in tool handle plastic to provide a weather proofing with only the wires exposed.

On the microcontroller side, I tied the sensors to external interrupts to enable counting of the turns of each wheel.

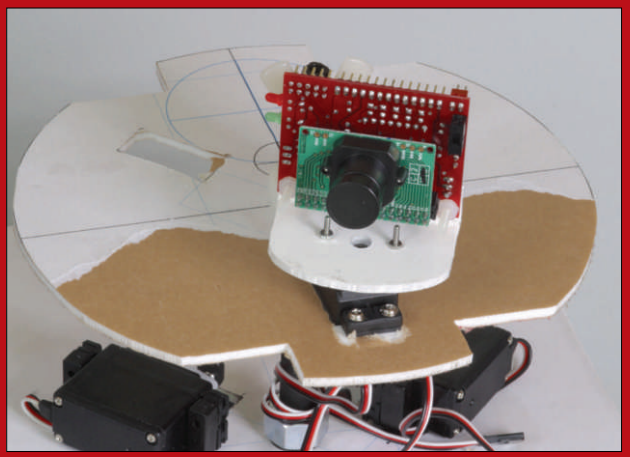


Figure 6. The CMUcam is shown mounted on a servo for panning capability. This servo is mounted to the head platform which can provide pitch and roll – in addition to its auto-level capability.

Since the microcontroller is also controlling what direction the bot is moving in, it then increments or decrements a counter for each wheel, thus eliminating the need to use quadrature encoding. With all this and knowing that each tire has a circumference of 31 cm, I am able to calculate the distance the bot has traveled.

In theory, calculating the distance should be simple, but — since each wheel can travel at different speeds — calculating the speed and distance the robot has traveled is more difficult. Although tire slippage is being minimized, it hasn't been completely eliminated. Since each wheel's rotation is being monitored, software is going to be used to "average" all the readings to determine how far the robot has actually traveled. "Average" is more of a generic term that means that all of the wheel movements are going to be analyzed to determine the bot's overall movement, which will be different, based on when the bot is going straight or moving through a turn.

One of the ways to confirm that the bot is traveling along the correct path is to use the various traffic cones that will be on the course, since their locations will be known. When a cone has been found and bumped, the bot will then know its exact location on the path. This is important, since any accumulated errors can be reset and the bot will be ready to resume its path to the final destination cone.

Lastly — for when all else fails — I included a GPS unit. I chose the Garmin GPS 18, an OEM unit that has a TTL serial interface and uses the latest technology. While I was hoping this unit would decrease the error margin that I normally see in other GPS units, I only experienced a minor improvement. Because of this error — on the order of eight meters — I relegated the GPS to be a last ditch method for finding a target cone when the dead reckoning navigation states I should be on location and the search patterns with the vision system (a CMUcam) have failed to find the cone.

Target Detection

The primary method my bot is using to "see" the target traffic cone and perform the final close proximity navigation to it uses the CMUcam. This CMUcam vision board provided a nice solution for this, since it can track a user-defined color blob and, thus, provide useful information that can be used to direct the bot to the cone. Although it can actively drive a servo to direct the camera to track a specific color blob, I decided to not use this onboard feature and provided my own.

The board requires a voltage greater than 5.5 V, but below

Talking to the CMUcam

There are many commands that the CMUcam has available, but I wanted to demonstrate the few that I used to handle all my needs. The commands all use visual ASCII to represent values and acknowledgements. There are a few that return binary data, but I don't have a need for them (yet).

All commands sent must end with a carriage return (ASCII 13). Upon a successful command transmission, the acknowledge string "ACK" is returned, followed by a carriage return. If a detectable error or unknown command was sent, the not-acknowledged string "NCK" is returned followed by a carriage return. When a carriage return is sent back followed by a colon, this means the camera is in a prompt mode and is awaiting another command.

When the camera first starts up, it will issue a prompt:

Received > :

Many commands will continue to send data back until they are stopped. This is done by issuing a break command:

Send >
Received > ACK
Received > :

In my application, I wanted to switch to CrYCb mode, but this same command is used to turn the auto white balance on and off. I use the Camera Register command:

Set Camera Register — CR [reg1 value1 [reg2 value2.. reg16 value16]]

Switch to CrYCb mode with white balance on

Send > CR 18 36
Received > ACK
Received > :

To switch to CrYCb with white balance off, the command is:

Send > CR 18 32
Received > ACK
Received > :

When I sample the center of the camera image for the color of the target cone, I set the camera to use a portion of the view (a window) and request it to sample the mean and deviation of colors.

Set Window Size — SW [x y x2 y2]

Get Mean color — GM

Data Packet S — S CRmean Ymean CBmean CRdeviation Ydeviation CBdeviation

Send > SW 30 30 50 113
Received > ACK
Received > :
Send > GM
Received > ACK
Received until stopped > S 40 80 90 8 3 4

When I track a cone, I reset the window that the camera uses and request it to track the color I define.

Track Color — TC [CRmin CRmax Ymin Ymax CBmin CBmax]

Data Packet M — M [MiddleMassX MiddleMassY UpperLeftX UpperLeftY LowerRightX LowerRightY Confidence

Send > SW
Received > ACK
Received > :
Send > TC 25 55 10 240 75 105
Received > ACK
Received until stopped > M 41 72 22 10 69 143 223

30 V, due to the onboard LM2940 voltage regulator; however, documentation suggests using between 6 and 7 volts. I will be providing 7.2 V from my batteries and the board should be able to handle that. It will draw 200 ma of current, but, since the camera is only needed part of the time, I added an external feature to remove power to the board on command.

The board communicates to the host processor through RS-232 or TTL serial communication. I used the TTL interface and left the interface configured to the default speed of 115,200 baud. Changing the speed has implications on the data format for some commands, but I was mainly interested in getting the data as fast as possible. The other communication parameters are static at eight data bits, one stop bit, no parity, and no flow control.

I mounted the camera on top of a servo to provide rotational scanning ability (Figure 6). With this, I can then scan the area within 120° in front of the bot without the need to turn the bot. Further, this system is mounted onto the same head mast as the compass sensor so that it is level at all times. If readings of the compass or other sensors that need to be level are not required, it can be used to tilt the camera.

When the camera is in tracking mode, it returns the xy middle mass coordinates of the color tracking blob, the bounding rectangle of the color tracking blob, and the confidence value of its tracking. The coordinates returned are in the range of 1 to 80 for x and 1 to 143 for y. These are used to "servo" the bot toward the cone. If the camera fails to lock onto a target cone, the bot simply makes a 90° turn and repeats the scan. If, after repeating the first scan, it doesn't find a cone, it will reevaluate its location and try to navigate closer to the cone before attempting another visual scan.

Like most cameras, there are many lighting issues that must be handled to get the best results. First, there is white balance. White balance is the adjustment to the image to account for the light's color. Sunlight has a different color tint than fluorescent light and, thus, the image is affected by the light source. Second, there is gain. Gain is the definition of the adjustment to the image to account for the light's brightness level. Clear skies will require a different gain than cloudy skies. Since this event is outdoors during the day, the white balance will not change as much from the start to finish as it could indoors. Reflected light from buildings, the time of day, and cloud layers could, however, affect white balance.

The documentation on the CMUcam suggests letting the camera auto white balance for a period of time before use and then turn it off. The reason for turning it off is that, if you pointed the camera at a green wall for a minute, it will state that the wall is white (or a shade of grey), as it constantly auto adjusts. However, if you point the camera arbitrarily and do this, the reading will be close, but still arbitrary, as it samples across the entire scene. Photographers have a solution to this problem. You can purchase a white card (or grey card) that you point the camera at under the selected lighting conditions to get the camera to accurately white balance. Since my camera can rotate and is on a tilting head, I mounted a card on the bot and, when I initialize the camera, I point it at this card and set the white balance.

The documentation further suggests letting the camera auto-gain for a period of time before use, then turning it off again. The reasoning is that the brightness level will affect the color tracking range and make it fail to see the color blob. This would be true if light were coming evenly from all directions, but — when light comes from one source, like the sun — there are going to be dark and light areas and looking at the object from the shadow side would also fail. I plan to leave the camera in auto-gain at all times and solve the issue of color tracking brightness values in a different way.

One of the commands to the CMUcam allows you to set the unit into CrYCb mode rather than the normal RGB mode. In CrYCb mode, the color components are separated from the brightness level component; therefore, brightness is defined by the Y value and the object color is defined by the Cr and Cb components. This gives me the benefit of providing our target's color range, clamped close in the color components, while being more flexible in the brightness component. This should help alleviate the problems of color tracking in varying light conditions.

Since I will be powering down the camera between target cones, I will need to set the white balance each time it powers up. What about defining the target colors, though? These values must be set before the contest attempt is started and there is no chance to set them afterward. I have a mode in my bot that allows me to teach it the color of the object. First, it powers up the camera and auto white balances it and, then, it centers the camera forward, expecting to see something in the center that represents the object. I just make sure a cone is present, centered in the view, and I give the camera the command to sample and store the data for later camera initialization.

Next Time

In this article, I discussed my bot's sensors and how they are being used. Next time, I will talk about the microcontroller and software on the bot, along with my waypoint tagging solution using a PDA. I will finish the series with some testing observations and the results of my entry in the Robo-Magellan contest. SV

Links

The Robson Company, Inc. — **www.robsonco.com**
Dinsmore analog compass sensor

CMUcam vision sensors
www.cs.cmu.edu/~cmucam

NVE Corp. — **www.robsonco.com**
AD221-00 magnetic switch

Garmin — **www.garmin.com/products/gps18oem**
Garmin GPS 18

New Products

ACCESSORIES

Seagull Wireless Telemetry System

The new Seagull Real Time Wireless Telemetry system is designed to transmit data in radio controlled model planes, cars, helicopters, boats, and other R/C models.

Features of the Seagull system include:

- 16 channels of telemetry data in real time
- Small, self-contained LCD Telemetry receiver simply attaches to your radio
- Powered from your receiver battery
- 65,536 unique IDs reduces interference
- Receiver is a USB peripheral; configuration is a snap on your PC
- Switchable low or high output power mode.
- Switchable sample rate for high resolution or longer battery life.

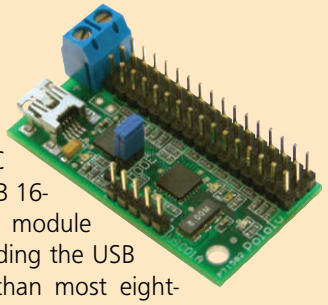
Depending on your application, some or all of these data can be transmitted to the Seagull Receiver: servo movements, receiver battery voltage, speed, altitude, RPM, and temperature. Optional expanders for measuring G-force, exhaust gas temperature, electric motor current/voltage, and other parameters are available from Eagle Tree Systems. All data from these expanders is compatible with Seagull.

For further information, please contact:

Eagle Tree Systems, LLC
4957 Lakemont Blvd. SE #C-4 PMB 235
Bellevue, WA 98006
Fax: **425•484•4131**
Email: **info@eagletreesystems.com**
Website: **www.eagletreesystems.com**

Circle #123 on the Reader Service Card.

USB 16-Servo Controller



Pololu announces a new addition to its line of R/C hobby servo controllers: the USB 16-Servo Controller. The compact module measures just 1.9" x 1.0" including the USB connector, making it smaller than most eight-channel controllers, and it offers both USB and asynchronous serial (UART) connectivity. No features are compromised for the small size, as the Pololu USB controller supports independent speed and range settings for each servo while delivering 0.5-microsecond resolution for smooth output across its broad output pulse range of 250 through 2750 microseconds.

When installed, the USB servo controller appears as a serial port to the host computer. Programming is therefore as easy as sending commands to a serial port and, as an added benefit, the servo controller is compatible with many existing programs. The servo controller can be connected to additional serial devices, such as Pololu's serial motor controllers, and thereby function as both a servo controller and a USB-to-UART bridge. The servo controller is compatible with USB 2.0 standards, and driver support will initially be available for Windows 98 through XP.

With its dual USB and UART interface, the servo controller can at first be used with a PC to quickly develop motion sequences with the advantage of graphical interfaces and quick program changes. For projects without dedicated PCs — such as small autonomous robots — the final motion sequences can then be transferred to an embedded controller, which can communicate with the servo controller over the standard asynchronous serial connection. With the trend toward removing serial ports from new computers, the Pololu USB 16-Servo Controller provides one of the most simple, small, and economical alternatives to standard serial servo controllers. The unit price is $81.00, with free shipping in the US. For further information, please contact:

Pololu Corporation
6000 S. Eastern Ave., Ste. 5-E
Las Vegas, NV 89119
Tel: **877•7•POLOLU** or **702•262•6648**
Fax: **702•262•6894**
Email: **www@pololu.com**
Website: **www.pololu.com**

Circle #77 on the Reader Service Card.

Micro 2.4 GHz Video Link — License Free

SUPERCIRCUITS of Liberty Hill, TX announces its new MVL10 micro-sized video link that has the robotic enthusiast in mind.

The MVL10 is a Part 15 FCC approved device for license free operation. With the built-in omni-directional antenna, a 700' line-of-sight range is conservative and will work with any video camera or microphone using a line level signal. The

MVL10's four-channel capability makes two-way audio, stereo vision, and operations in RF-unfriendly environments possible.

The MVL10 transmitter has a small power requirement of 100 milliamps at 12 volts. The transmitter weighs a light 0.8 ounces and the board size of this feature-packed unit is a thin 0.42 inches and it is 2.2" squared.

For further information, please contact:

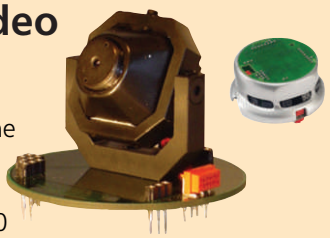
Supercircuits

One Supercircuits Plaza
Liberty Hill, TX 78642
Tel: **800•335•9777** Fax: **866•267•9777**
Website: **www.supercircuits.com**

Circle #140 on the Reader Service Card.

CONTROLLERS & PROCESSORS

New Wireless Video Camera Turret



For those unfamiliar with the Khepera, it is a miniature robot — smaller than a hockey puck, weighing only 80 grams, and sporting a Motorola 68331, 25 MHz processor with 512 Kbytes RAM, and 512 Kbytes flash. It is used by robotics researchers world wide.

The new Wireless Camera Vision Turret features a CMOS 450 x 380 pixel color vision sensor and an embedded microphone. The video and audio are transmitted on one of four channels in the 2.4 GHz band. It is designed to transmit sound and video from the Khepera to a base receiver connected to a computer. The sound and images can then be processed on the remote computer, followed by serial commands sent back to the Khepera through its Radio Turret extension.

The camera is very easy to connect and use; no settings or software are required. The camera orientation can be changed to match specific experimental needs. The base receiver has simple cinch video and audio outputs that can be connected to a video frame grabber or to an audio computer input. It comes with an external power supply.

The camera's maximum scan frequency is 50 Hz and its maximum distance is 100 meters. The output power is 10 mW and it operates with a minimum illumination of 1 lux.

Khepera and its accessories are designed and manufactured by K-Team, S.A. (Lausanne, Switzerland).

For further information, please contact:

RoadNarrows, LLC

1151 Eagle Dr. #140
Loveland, CO 80537
Email: **oneway@roadnarrowsrobotics.com**
Website: **www.roadnarrowsrobotics.com**

Circle #27 on the Reader Service Card.

Casting and Mold Making

by Gordon McComb

Casting and mold making aren't skills shared by most robot builders, but they should be. With little training and only a few dollars in materials, you can produce your own parts for your robo-creations. The process is simple enough: Use plaster or some other material to create a mold from an existing element.

One possible use is hubs for wheels. As long as you have one hub for the model, you can make additional ones as needed. The hub can be something you've purchased, retrofitted from some other item, or custom made using wood, metal, or plastic.

After the mold has dried and cured, you can then cast parts from it. Castings can be made using hard or soft rubber, even certain low temperature metals.

The benefits of casting are twofold. First, you can save money by making your own parts. Buying new sprockets, gears, wheel hubs, and other components can get expensive. Once you make a mold, each casting might cost only 25 cents each, depending on the material — obviously, some casting materials are more expensive than others — but, overall, it's pretty inexpensive stuff.

Second, you can "mass produce" unique items, rather than hand-make them, one at a time. You only need to build one model, then make duplicates with casting.

Casting in Plastic

Casting an object in plastic is not difficult, but it can be time consuming. The general steps are: create the mold, then create the casting. There's a bit more to it than that, of course. Here's a very quick overview:

Create the Mold

All castings require a mold of some type. If a mold doesn't yet exist, you will need to make one from a model. The most common molds are made of plaster, which you can purchase at any arts and crafts store. Specialty plasters, such as dental plaster (examples: Hydrocal and Die-Keen), are available from mail order outlets. The mold is produced by glopping the semi-liquid plaster over the model and letting the plaster dry.

Plaster molds are either one piece or two piece. A one piece mold has just one half and is suited for objects where only the front side needs to be cast because the final object is meant to be placed against a flat panel (examples are decorative trims or some kind of flat hood ornament for your robot). A two piece mold has a front and a back and is meant to cast a standalone figure.

Other mold materials exist, of course, and are better suited than plaster for some tasks. Plaster molds can't have any "undercuts" because the plaster dries hard and stiff. The more irregularly shaped the model is, the more you will need to use a flexible mold material, such as latex. Latex molds do not hold their shape for the casting step, however, and need to be reenforced with a plaster "mother mold." In all, making latex and other flexible molds takes extra time.

Create the Casting

A casting is made by pouring a liquid or semi-liquid material into the mold and letting the material set up and cure. Some casting materials, like hot melt (very much like hot glue), set up when they cool. These are the fastest to work with. Others, like urethane resin, require a chemical change that can take several hours or even overnight.

Most liquid plastics (like urethane) for casting are composed of two parts: the main plastic resin and a catalyst. Separately, the materials will never harden to produce a finished plastic piece. When mixed, a chemical reaction causes curing and results in hardening. The catalyst must be added to the plastic in just the right amounts or

the casting will be ruined. The mixture is either determined by weight or by volume; each manufacturer will provide an instruction sheet that you should follow to the letter.

Depending on the materials you use, you may need to apply what's known as mold release before pouring in the liquid casting gloop. Petroleum jelly is common for plaster molds, though there are also thinner sprays available that create less mess. It is critical that the mold release be compatible with both the mold and casting materials or else the casting may never fully cure.

The single greatest error made by beginners to the mold making and casting art is trying to rush things. It can take a day or two between start and finish to complete a casting, especially if a mold is also required.

Also, depending on the casting materials you use, a plaster mold will need extra time — days or even weeks — for the water in the plaster to dry out. (However, some moisture is good, as it keeps the mold from becoming too brittle. You may actually have to add water to your molds periodically, especially if you cast with urethane resins, which draw out moisture as they cure.)

Trying to rush things will result in a poor casting that either never cures (remains soft and sticky), comes out of the mold in pieces, or is too brittle. If you take your time, you'll find that your casting results will be up there with the pros.

Safety and Respirator Masks

Materials used for mold making and casting vary from harmless dust producers to downright carcinogens. You must exercise care when working with mold making and casting materials and remember to wear a respirator mask, safety goggles, and — if necessary — gloves.

Plaster for molds presents little danger, except that you should never use it to make a mold of an arm, leg, or other body part (yours or someone else's). Plaster cures by producing heat and this heat can burn skin. Urethane and fiberglass resins should always be used in a well-ventilated area where the fumes and vapors can escape out into the rest of the world. Never try to cast plastic in an enclosed room and advise others that they may need to leave for a while.

Smoking is a no-no while working with molds and casting; also, keep all ingredients away from any open flame. Some casting materials require heating (this is the case with hot melt); use an electric double-boiler or hot plate to heat the material. Don't use an oven or range that uses gas and produces an open flame.

Selecting the Right Plastic Casting Resin

There are plenty of plastic casting resins to choose from. The more you cast your own parts, the more you'll discover which casting resins are the most suited for the work you do. (You'll also learn such important things as which ones stink less.)

See the table for a short run-down of several popular products and manufacturers. Browse each company's website to discover the ancillary products that might be of use to you, such as curing additives (these slow down or speed up curing), colorants, additives to change the texture and hardness of the final casting, and much more.

Using Fiberglass Casting Tape

If you've ever broken an arm or leg, the doctor probably gave you a fiberglass cast. Rather than get out sheets of fiberglass matting and mixing resin and catalyst in a big jug, the doctor likely just ripped open a package of pre-soaked fiberglass tape and started wrapping it around your body. The material cures in a matter of minutes.

The benefit of using pre-soaked fiberglass tape is obvious and ease-of-use is at the top of the list. The same casting tape doctors use is available to mere mortals. You can use it to build robust robot bodies and even repair ones that get damaged.

3M Scotchcast is a popular casting tape available from medical supply companies. The tape is simply dipped in water to activate it, can be cut with scissors (before or after being dipped into water), and can be worked into the desired shape over a period of several minutes before it begins to harden. Casting gloves are recommended because the tape gets sticky and the urethane resin is a skin irritant. The tape is available in widths from 2 to 5 inches.

Scotchcast is made for both human and animal use. Veterinary casting supplies are often cheaper. Try the following **Google.com** searches to locate suppliers of casting tape:

- *veterinary supplies + equipment*
- *casting tape*
- *orthotics supply*

Hot and Cold Dipping Plastics

Dipping plastics are like fondue with polymers. Perhaps the best known application for dipping plastics is applying a kind of rubberized grip to hand tools — just submerge the handle of the tool in a can of dip, pull it out after

Product	For more information, visit
Alumilite	www.alumilite.com
Armorcast	www.armorcast.com
Fast Cast	www.goldenwestmfg.com
Foam latex	www.burmanfoam.com
Polyurethane plastics	www.polytek.com
POR-A-KAST	www.synair.com
Urethane casting resins	www.smooth-on.com

a few seconds, and let it dry. Most dip can also be applied by brushing, but the idea is the same.

Dipping plastics harden into a tough shell or a rubbery sheath, depending on the chemistry used. There are two basic flavors: cold and hot. Cold dip is not heated before use and cures with exposure to air. Hot dip is cured by warming it in an electric oven, as per the manufacturer's directions (typically 350-375° for 15 or 20 minutes). The plastic completes its curing process when cooled. For robotics, dipping plastic can be used to add a rubberized soft coating to the legs of walking bots, to frame bumpers for damage reduction, to add traction to wheels, and even to provide extra insulation for wire connectors.

Here are some tips for using dipping plastics:

• Curing times for cold can be reduced by gently warming with a hair dryer.

• Apply several thin coats rather than one thick coat.

• Hot dipping is best when the object being dipped is pre-heated. Obviously, this won't work if you're using plastic or wood. It works well for metal objects, like the aluminum legs on a walking robot.

The most popular dipping material is plastisol — a PVC compound that can be made in most any color, thickness, and hardness (when cured). In addition to being used as a heavy coating, plastisol serves as a semi-flexible mold.

Casting in Metal

A far older technique than casting in plastic is casting in metal. Metal casting is found everywhere and is used to produce both small parts and large ones. While metal casting in commonplace in industry, it's relatively rare in amateur robotics. First, cast metal tends to be quite heavy; most amateur robots are small, so it's important to keep the weight down.

Second, like plastic casting, metal casting requires a mold. The mold must withstand the high temperatures of molten metal, so making the mold is far more involved. The typical approach is to machine a mold using a steel alloy that has a higher melting temperature than the metal that will be cast. Other mold materials include sand (which can be more readily formed into a shape), investment (lost wax), ceramic, and plaster. The softer the mold, the fewer casts it will make. Some molds last for only one cast.

Sand casting is one of the most useful techniques for producing limited-use molds. There are several variations:

• *Green sand* molds are made by packing loose sand around a pattern and removing the pattern to form a hollow mold.

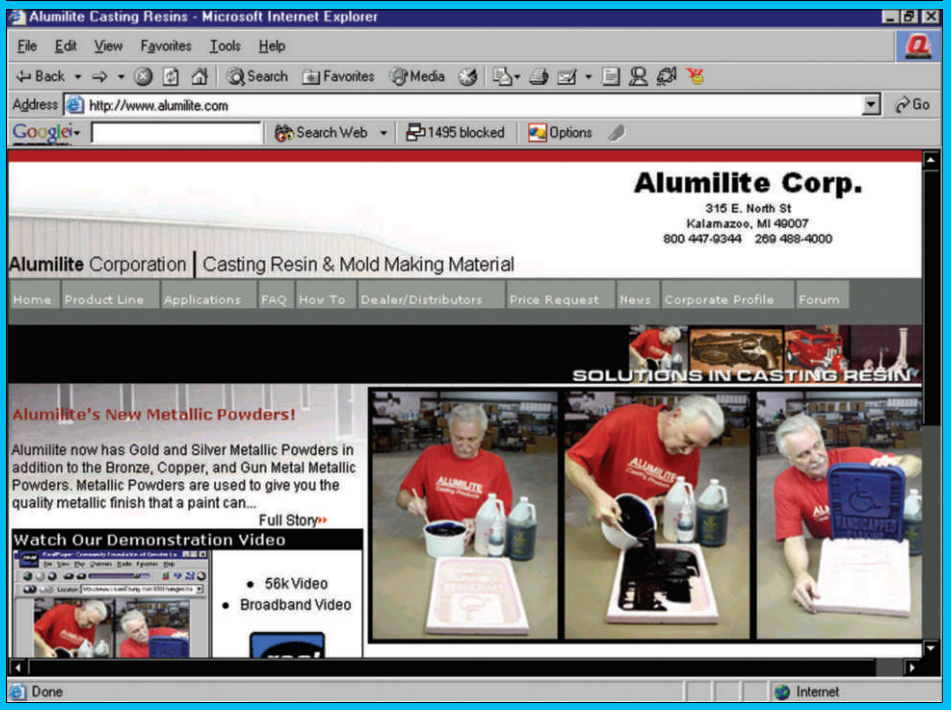
• *Shell molds* are made by heating a resin-sand mixture in metal frames, forming top and bottom shells that are joined to make a complete form. The benefit of the shell mold is that the resin hardens and helps the sand keep its shape.

Sand casting uses disposable molds, but — because the mold making process is relatively quick and easy — several molds can be economically produced from a single original for "production work." The original can be made out of wood, plastic, plaster, or almost any other material. A variety of metals can be cast using the sand methods, including iron, bronze, and aluminum.

A relatively new technique of producing finished metal pieces uses a material generically known as precious metal clay. Though intended for use in making jewelry, this material is also suitable for a number of casting applications. Precious metal clay (or PMC) is a moldable clay that is impregnated with a fine powder of silver or gold and sometimes other metals, as well. When heated ("fired"), the clay burns away, leaving the molded metal object. The best results are obtained using a kiln that can generate in excess of 1,500° F. Such kilns are available for the hobbyist at prices starting at around $300.00.

One use of PMC is to produce molds for plastic or metal casting. You can make single or two-piece molds in

FIGURE 1. Alumilite casting resins, foams, and casting supplies.



this manner. The process is ideal for parts that require a fair amount of detail. The clay can be worked with your fingers or ordinary clay sculpting tools. It will also take the impression of a 3-D object, such as a gear or sprocket.

Casting and Mold Making Sources

Abatron, Inc.
www.abatron.com

Abatron is a manufacturer and distributor of adhesives, sealants, coatings, epoxy, polyurethane, and various other resins. Of particular interest to robot builders is their casting and mold making supplies.

One interesting product is Woodcast, a two-part, light casting material that is machinable and even stainable. You basically mix up the liquid and pour it into a mold. You can shape and sand the material afterward. If the "natural look" isn't right for your robot, there's also Abocast — a two-part clear liquid for casting. Abocast is a tooling resin, so you can use it to make complex shapes, should you have access to a mill, lathe, or computerized router.

Alumilite Corporation
www.alumilite.com

Alumilite sells casting resins and mold making materials. The company sells many supplies in small quantities or in bulk, as well as the Super Casting Kit and Mini Super Casting Kit — both designed for the consumer market. Additional products include:

- Urethane dyes
- Metallic powder
- Microballoons
- Alumilite 610 Foam (expands to 6-10 times the original liquid volume)

The website includes numerous how-tos on mold making and casting.

American Art and Clay Co., Inc. (Amaco)
www.amaco.com

Amaco manufactures and sells (principally through retailers) a line of arts and crafts products which include WireForm wire mesh (can be sculpted to form a shape), Friendly Plastic (softens in warm water), Easy Metal (thin metal foil), and glow-in-the-dark paints.

Armorcast
www.armorcast.com

This site offers polyurethane resin molds. Much of Armorcast's product line is "scenery" — things like miniature trees and rocks for models that are suitable for robot display or for a playing field for your combat bot.

Bare-Metal Foil Co.
www.bare-metal.com

Bare-Metal Foil sells bare metal foils, silicone RTV (which stands for room-temperature vulcanizing), casting resins, decals, and other products for the model maker. Useful how-to articles are included.

Burman Industries, Inc.
www.burmanfoam.com

Burman is in the business of monsters. Making them, that is. They sell foam, latex, and other materials for special effects make-up, mold/model making, puppetry, and animatronics professionals. Their adhesives, clays, latex, mold making products, tubing, armature wire, aluminum pieces, and other products can be used to make robots — monstrous or not.

Castcraft
www.castcraft.com

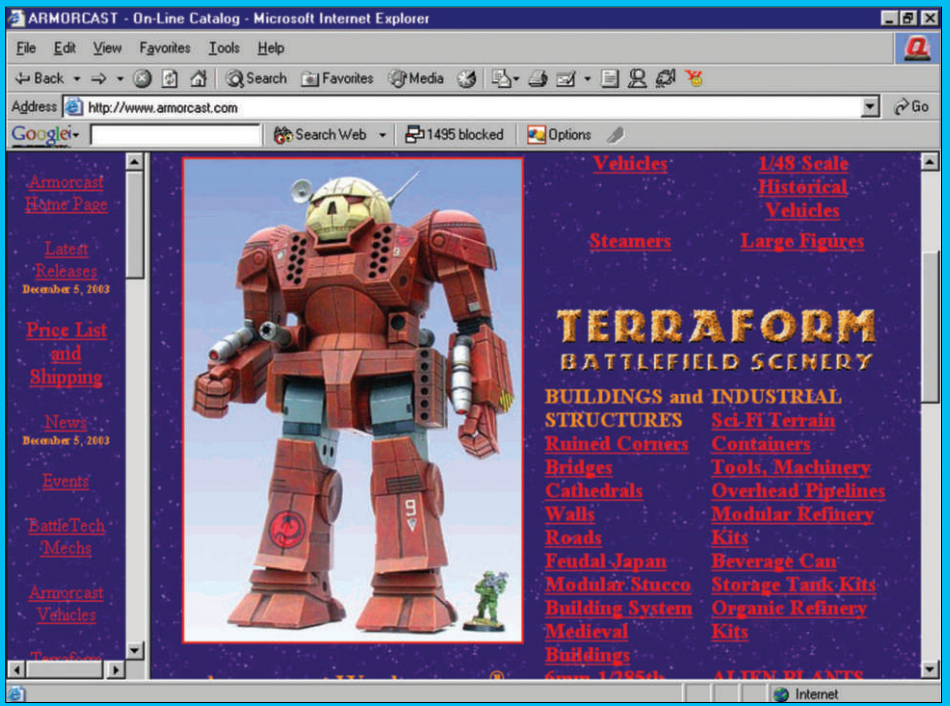
Castcraft calls itself "the place for mold making and casting." Along with metal, plastic, and plaster casting, they offer products, plans, and parts to build a homemade vacuum-forming machine.

Douglas and Sturgess, Inc.
www.artstuf.com

Here, you'll find casting and mold making supplies for the sculptor: sculpture tools, materials, and supplies. Much of it is also useful for most any casting and mold making project that might involve robots or robot parts. Interesting products include:

- Atomized metal powders
- Coating materials
- Latexes and other flexible mold materials

FIGURE 2. Armorcast mold supplies and resin scenery (for dressing up your robot).



- Foam board materials (such as pink styrene and polyurethane)
- Fabricform (plastic laminated between pieces of fabric)
- Altaform (general use substrate in three different weights)
- FormFast (also known as celastic): plastic-impregnated fabric softens with acetone and can be shaped and left to re-harden

Environmental Technology, Inc.
www.eti-usa.com

This company is a manufacturer of casting and molding supplies: epoxies, adhesives, casting compounds, latex rubber, etc. The products are available in industrial packaging. Contact the company for more information. The Castin' Craft brand — available in smaller quantities — can be purchased at local craft stores.

eWellness
www.ewellness.com

This is where you'll find medical supplies, including fiberglass casting supplies — you use it to make stuff for robot bodies, of course, not fix broken arms and legs. The site is for consumers.

Far West Materials
www.farwestmaterials.com

This is a source for mold making and casting supplies. The company specializes in the distribution of foundry, sculpture, and molding materials.

Fibre Glast Developments Corporation
www.fibreglast.com

This manufacturer specializes in fiber glass and laminates. According to the company, Fibre Glast is the world's leading supplier of composite materials for industrial, commercial, and hobbyist applications.

Freeman Manufacturing & Supply Co.
www.freemansupply.com

Freeman sells materials for machine tooling, including machinable wax — a kind of hard wax that is used when creating prototypes using a mill or lathe. The wax melts at about 225° and can be "cast" in almost any shape. It can then be routed, milled, or lathed into complex machine parts.

GoldenWest Manufacturing
www.goldenwestmfg.com

GoldenWest manufactures casting resins (both rigid and flexible), machinable plastic, foam board, and tools for cast and mold making. Their machinable plastic includes a product known as Butter-Board — a lightweight plastic block that is non-abrasive and very easy to machine or work with hand tools.

Life-casting.com/Artmolds
www.artmolds.com

This molding and casting supply dealer emphasizes life-casting (heads, bodies, hands), but the materials can be used for other projects, including almost-alive robots.

Michael Burnett Productions, Inc.
www.mbpfx.com

If you look through the movie hype, you'll find make-up and special effects products, including casting resins and latex.

Miniature Molds.com
www.miniaturemolds.com

Metal casting kits, supplies, and instruction can be found here. They sell Alumilite, RTV, various metals and rubber, and molds. The company specializes in molds and kits to make small character figures, but, of course, the same materials can be used for small character robots.

Moore Medical
www.mooremedical.com

Moore sells medical supplies; you're interested in fiberglass casting tape and plaster casting materials.

Perma-Flex Mold Co., Inc.
www.perma-flex.com

Perma-Flex is a supplier of rubber casting and mold making products. For example, their polysulfide rubber compound creates flexible molds (good for irregularly shaped objects) and can withstand multiple castings. It is sold in quantities ranging from one quart to a 600 pound drum. Some basic tutorial and reference material is provided on the site.

Plaster Master Industries
www.plastermaster.com

Casting and mold making supplies, including gypsum plaster, foam, resin and casting materials, and mold and model making supplies are here.

Polytek Development Corp.
www.polytek.com

Polytek manufactures flexible molding and resin casting materials, including RTV silicone mold rubber, polyurethane plastics, polyurethane foam, and release agents.

Schenz Theatrical Supply, Inc.
www.schenz.com

Schenz specializes in products for high school and community theaters. Products include pre-made props (armor, masks, swords — that sort of thing), but they also offer some inexpensive casting and mold making supplies, such as Celastic, liquid latex rubber, and metallic powders.

Sculpture House Casting
www.sculptshop.com

Sculpture House Casting sells casting and mold making supplies and tools, armatures, and modeling accessories.

Smooth-On
www.smooth-on.com

Smooth-on is liquid rubber and plastic. With this stuff, you can cast or coat items in plastic and rubber by either pouring, spraying, or brushing on. The site contains products (including samplers), how-to, and a newsletter. Among the most useful of casting materials is rigid urethane resin, which you can use to make your special parts, including sprockets and gears.

Special Effect Supply Co.
www.fxsupply.com

Where do I start? Okay, how about these goodies for movie and

stage special effects, which, of course, are also useful for robotics:

- Eyeballs
- Hemispheres (clear acrylic domes, diameters to 6 inches)
- Latex rubber
- Make-up prosthetics
- Molding and casting
- Plaster and gypsum
- Pneumatics (solenoid valves, cylinders, hose — everything but the air)
- Sculpting clay
- Sculpting stuff
- Silicone kits and materials
- Skeletons and corpses (*Tales of the Crypt* robots?)
- UV and DayGlow makeup, paint, and material
- Vacuum forming

Look well beyond the box for some of these products and you'll think of some interesting and useful ideas. Most product categories include basic background information.

Syn-Air Corporation
www.synair.com/mold_ind/

Syn-Air makes mold making products. Offerings include Por-A-Mold (polyurethane rubber, cures at room temperature), Por-A-Kast (liquid urethane casting compound), and Sil-Mond (two component, cures at room temperature). They also make polyurethane based in-mold paints, adhesives, binders, and release agents.

The Compleat Sculptor, Inc.
www.sculpt.com

Materials, supplies, tools, and service for sculpting, mold making, and casting are here.

Urethane Supply Company
www.urethanesupply.com

This is a site for plastic welders and supplies, mainly for automotive and boat repair. Good technical information and a regular newsletter in Adobe Acrobat PDF format are offered.

Van Aken International, Inc.
www.vanaken.com

Van Aken is a manufacturer of art supplies. Their products include the famous Tempera and Plastelene. The Van Aken site has some useful semi-technical backgrounds and application ideas. The product is available through retailers. **SV**

About the Author

Gordon McComb is the author of the best sellers *Robot Builder's Bonanza*, *Robot Builder's Sourcebook*, and *Constructing Robot Bases*, all from Tab/McGraw-Hill. In addition to writing books, he operates a small manufacturing company dedicated to low cost amateur robotics, **www.budgetrobotics.com** He can be reached at robots@robotoid.com

EVENTS CALENDAR

Send updates, new listings, corrections, complaints, and suggestions to: **steve@ncc.com** or FAX **972-404-0269**

There are quite a few competitions being held in the last third of 2004. They range from the traditional — soccer, line following, maze solving, sumo and BEAM — to the esoteric, like bridge building in ABU Robocon.

Echoing Steven's comments from last month, the number of DARPA Grand Challenge clones is on the rise, as well. I am glad to see this, as I expect there to be a large demand for sensor fusion systems on AUVs in the near future.

One final thought: even if you don't think your skills are up to the level of entering these competitions, you should try to attend them. You might be surprised at the encouragement you'll receive just by standing in the audience — it's contagious!

— Editor Dan

For last minute updates and changes, you can always find the most recent version of the complete Robot Competition FAQ at Robots.net: **http://robots.net/rcfaq.html**

September

3-6 **Dragon*Con Robot Battles**
Atlanta, GA
Radio-controlled vehicles destroy each other at a famous science fiction convention.
www.dragoncon.org/

6-7 **RoboCup Junior Australia**
Queensland, Australia
There are over 600 RoboCup Junior teams in Australia. Regionals narrow this number down to about 200 teams that will compete at the University of Queensland to see who's the best at building LEGO-based, autonomous soccer robots.
www.robocupjunior.org.au/

11 **ABU Robocon**
Seoul, Korea
Autonomous robots must build a bridge and then move objects across it.
www.kbs.co.kr/aburobocon2004/

18 **DPRG RoboRama 2004**
The Science Place, Dallas, TX
Robots from the lone star state compete in the usual events: Line Following, Quick-Trip, T-Time, and Can-Can.
www.dprg.org/

25-26 **Robothon**
Seattle Center, Seattle, WA
At this Seattle Robotics Society event, people from all around the world come together to present new robotic technologies, show off their special robotic creations, and compete in several robotic competitions and activities. The Robo-Magellan competition is sponsored by *SERVO Magazine*.
www.robothon.org/

October

1 **IRRF Open Challenge**
Las Vegas, NV
Commercial clone of DARPA's Grand Challenge. Can private enterprise do it better?
www.irrf.org/

2 **Robot — Liga**
Kaiserslautern, Germany
Several events. If I could read German I'd tell you more about them ...
www.robotliga.de/

8-10 **Robot Fighting League National**
Herbst Pavilion, Fort Mason Center
San Francisco, CA
Radio-controlled vehicles destroy each other — and bank accounts — in San Francisco.
www.botleague.com/

9-10 **RoboMaxx**
Grants Pass, OR
Includes a range of events for autonomous robots, including maze solving, 3 kg sumo, mini sumo, micro sumo, and nano sumo.
www.sorobotics.org/RoboMaxx/

21-23 **Tetsujin**
RoboNexus, Santa Clara, CA
SERVO Magazine's weight lifting competition for powered, articulated exoskeletons offers an event incorporating the technology of the future. The event is being held in conjunction with RoboNexus. See page 8 of this issue for more information or visit the website for rules and full details.
www.servomagazine.com/tetsujin2004/

22-24 **Critter Crunch**
MileHicon, Marriott Southeast, Denver, CO
The Denver Area Mad Scientists were pitting autonomous and remote-controlled robots against each other long before commercial events like "BattleBots" and "Robot Wars."
www.milehicon.org/

27-31 **FIRA Robot World Cup**
BEXCO, Busan, Korea
Check out all the usual categories of robot soccer, including humanoid, single, team, khepera, and many others. Visit the website for further details.
www.fira.net/

November

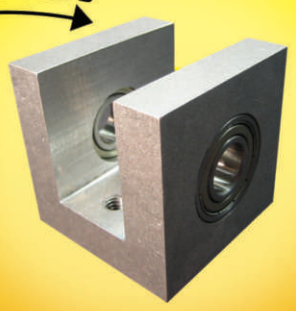
6 **CIRC Autonomous Robot Sumo Competition**
Peoria, IL
In addition to sumo, this year's event includes some R/C combat events.
www.circ.mtco.com/competitions/2004/menu.htm

10 **AESS National Robotics Contest**
Barcelona, Spain
Based on the photos available at the website, this is a student contest for line-followers and sumo robots.
http://aess.upc.es/concursrobot/

13-14 **Eastern Canadian Robot Games**
Ontario Science Centre, Ontario, Canada
Includes BEAM events, including autonomous sumo and a fire fighting competition.
www.robotgames.ca/

22 **Texas BEST Competition**
Reed Arena, Texas A & M University
College Station, TX
This is the big one, where the winners from the regionals compete.
www.texasbest.org/

26-27 **War-Bots Xtreme**
Saskatoon Saskatchewan, Canada
Robots (R/C vehicles) attempt to destroy each other to win a whopping $10,000.00 in prize money.
www.warbotsxtreme.com/

TEAM DELTA
ENGINEERED ROBOT PARTS SINCE 1999
ROLL ALONG!

RCM200
Dual Bearing Unit
$19.95
This 8.2 oz U-shaped aluminum pillow block is the perfect way to mount a drive axle, shoulder joint or waist on your new robot. 1/2" ID bearings and 1/4-20 threaded mounts - all inside a 2" cube!
SPECIAL FOR SERVO READERS → VISIT WWW.TEAMDELTA.COM/SERVO

I, Robot — A Movie Whose Time Has Come

by Dr. Joanne Pransky

I felt the hairs on my arm raise as I watched *I, Robot*. (I wonder ... if my arm was electro-mechanically modified like Spooner's, would I still get goose bumps?) I have had two other arm "hair-raising" experiences in my life: when I met Isaac Asimov himself and when I met Honda's Asimo. (Seems there's an "Asimovian" fixation going on here.) I don't usually encounter these experiences in my everyday robotic psychiatrist's life, but they're worth the wait. In fact, the wait was nearly 20 years before the issues that I'd been thinking about became water cooler talk overnight — after the release of the film, *I, Robot*.

Back in the mid 1980s, I envisioned stores where people could buy clothes and accessories for their robotic assistants like they do now for their pets. I also thought — as funny as it sounds — that one day we'd pay for that extra seat on the airplane to have our robots sit next to us. We'd legally fight about their custody in a divorce suit. We'd write our wills to provide for our bots' necessary future maintenance and upgrades long after our own deaths. When our robots were depressed (and "we consider" is the operative here), we'd take them to a robotic psychiatrist for their weekly sessions.

In 1986, I dubbed myself the "World's First Robotic Psychiatrist" and the real "Susan Calvin." This wasn't exactly one of the top 10 growth careers or even a formal course of study back then. When asked what I did for a living, I responded with "robotics."

Most people thought I said aerobics and I'd field questions on the latest Jane Fonda and Richard Simmons videotapes. It's 18 years later and some folks (i.e., mainstream America) still think I mean "aerobics." They obviously have not seen *I, Robot*.

I have to admit – I wasn't expecting much from the movie. I figured it would be another stereotypical Hollywood action flick like *Terminator 3*. Using the draw of a big star like Will Smith and some good special effects, it would be nothing like Asimov's stories and nothing that enlightened me. Susan Calvin is being portrayed by Bridget Moynahan? Susan Calvin would be 53 years old in the year 2035 — a role more suitable for Kathy Bates or an older, anorexic Calista Flockhart.

However *I, Robot* was more than surprising. I thought it was awe-inspiring — a movie whose time has come. I also thought its robotic technology was outstanding.

From the time the first NS robot is shown as a FedEx deliverer (personally, I would have thought the USPS could benefit more from efficient, automated carriers, but would we, as taxpayers, pay the film licensing fees?), I recognized that these are the types of robotic assistants I have been touting for two decades.

These are personal robots that do the mundane, difficult, repetitive tasks and are useful to us in our everyday lives: robotic dog walkers, prep cooks, bartenders, sanitation workers, "shleppers" to carry our groceries and other personal items, nannies, and aides that retrieve our medicine when we forget it and administer it for us when we're sick. Here is a robot in every home, one robot for every five people. Of course, this is nothing new for *SERVO Magazine* readers. Even in my home today, there's a ratio of five robots/human — if you include the defunct ones.

Asimov's Nestor series was two-legged. I've always thought that this form factor — despite the technical challenges of bipedal robots — would be best suited to accompany and work directly with us. I also felt the rest of the robotic embodiment was cleverly designed. The robots were easily discernible from humans, yet anthropomorphic enough to have features that humans are comfortable with: eyes, mouth, and facial expressions. The Nestor series physically represents what research has shown we want — a robot that looks like us, but not *too* much like us (the Uncanny Valley Theory).

Then there were several other robotic concepts — though based on existing service robot applications — that took us one more robo-evolutionary step into the future that we can look forward to: more robot artists (one of the first was Harold Cohen's "Aaron"), robots building and assembling 100% of the components of other robots, a robo-parking garage that rotates a car 180° before parking it, Survival Research Labs (SRL) making money by manufacturing demolition robots, robot fighting events that pit autonomous robots against other autonomous robots (and/or humans), and some team winning the DARPA Grand Challenge and having Audi purchase the technology.

I did question why there weren't

security robots instead of human police officers; even if a few were slightly 'RoboCop'-like in that they were cybernetically modified. Was it because it just didn't fit the whole storyline? Or is it that human cops are able to make better decisions than robots — such as in the case of the NS-4 who should have — but didn't — save the girl from drowning?

Was it a mere mathematical calculation of deciding to save Spooner because his chances of survival were 45% as compared to the 12-year-old girl's 11%? Should its decision be based upon the priority to save women and children first, regardless of the odds? Or was the robot faced with a decision of which life would be best saved for the benefit of other humans?

In other words, was the detective a higher priority to society than a young teen with mere aspirations and dreams and, therefore, was Spooner more useful in protecting other humans from harm?

These questions are just some of those raised by the interpretation of the famous Asimov's Three Laws — the raison d'être for this movie, as well as the stories in *I, Robot*. Though the movie did not emulate any specific story from the nine in the novel *I, Robot*, the credits at the end of the film clearly state that the film's storyline "was suggested" by the book, not based upon it.

The movie most closely resembled the 1947 story "Little Lost Robot," in which a Nestor (NS) robot is the villain. He was purposely constructed with a modified First Law, was berated, and told to "get lost," which it promptly did among the other 62 robots. In order to find the lost robot, Calvin and her team interview the robots and set up performance tests directly in response to the First Law.

"Little Lost Robot" and the other eight stories in the book address the analysis of the ingrained Three Laws. They also question whether there are circumstances under which a robot may need to lie. Under what acceptable conditions could a robot allow a human to come to harm? When might it be critical for a robot not to obey a human?

Though there was no Del Spooner in Asimov's stories, the detective's sentiments towards robots represent society's longstanding view — that of mistrust and dislike for robots. Asimov knew this would always be of great concern and, in "Little Lost Robot," he makes a reference to the "Frankenstein Complex." This was a term he coined to refer to the fear that machines will takeover and make us obsolete. This was reinforced by other science fiction writers — like Mary Shelley and Capek — as well.

The greater the autonomy and independence given to machines, the more frightening they are to humans, since a human's control decreases. Asimov knew the interrelationship of man (Spooner) and robot was not insignificant. Mankind may intellectually know about the Three Laws, yet humans will always fear and distrust robots.

Moynahan's portrayal of Dr. Susan Calvin was quite perceptive. She was colorless, serious, and stiff, but her passion, commitment, and compassion for robots was evident. Indeed, it was much deeper than her experiences with humans were. Regardless of her conservative clothing, her plain features, and asexual behavior, Moynahan was still physically appealing. She certainly did not resemble Asimov's gray-haired, unattractive Calvin. Perhaps Asimov purposely made Calvin this way, since he loved women so much and would find her to be a distraction if she were beautiful.

More precisely, if Calvin in the novel had been attractive, she may have not been as dedicated to her work. She may have ended up married, with children of her own, or — at the very least — with some sort of social life (though it could also be argued that none of those would have transpired, since Calvin hated human beings).

It has also been written that Asimov based Calvin on Rear Admiral Grace Hopper — a brilliant scientist who received her doctorate in mathematics from Yale in 1934 and was later credited with inventing the compiler and coining the term "computer bug." Not only was it rare for a woman to have been so well educated during that era, but, also having excelled in an all-male profession truly made Hopper a pioneer.

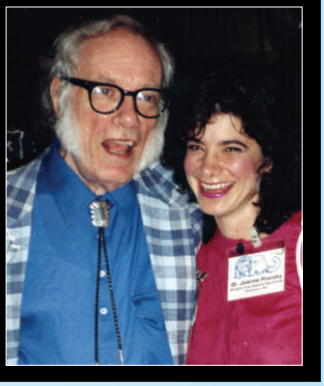
For Asimov in 1940 to have chosen a woman to be the robopsychologist was quite a compliment to the gender. Regardless of whether he did so with Hopper in mind, he must have thought that a female would be the intuitive gender better suited to nurture, empathize with, and understand robots.

I think Asimov would have applauded the sensate and emotional Sonny. Was that name chosen because of the word "son" in it (referring to Lanning's son) or because Asimov's robot names were similar to their model names? Thus, the "SN" in Sonny becomes the reverse of the "NS" series for Nestor.

Sonny depicted the intelligent and complex robots Asimov wrote about in

Author Bio: She's the *Real* Susan Calvin!

Dubbed the "Real Susan Calvin" by Asimov himself in 1989, Dr. Joanne Pransky, the World's First Robotic Psychiatrist®, brings Calvin to life in a tongue-in-cheek way to prepare the public for a future when robot companions are a part of everyday life.



Pransky has been a marketing pioneer in the robotics industry for nearly 20 years. She was the senior sales and marketing executive for a major industrial robot manufacturer and is currently an associate editor for *Industrial Robot Journal*, Director of Marketing for the newly launched *International Journal of Medical Robotics + Computer Assisted Surgery* (**www.roboticpublications.com**), and a consultant sought out by service and personal robot companies.

Visit her on the Internet at **www.robot.md**

his essays in the late '70s and '80s. As a matter of fact, the scene in which Spooner questions Sonny on a robot's ability to create art and music is based on the following written words of Asimov: "Some people are sure to be disbelieving and say, 'But how can a computer possibly produce a great symphony, a great work of art, a great new scientific theory?' The retort I am usually tempted to make to this question is 'Can you?'"

VIKI (Virtual Interactive Kinematic Intelligence) symbolizes Asimov's addition of the Zeroth Law. In 1985, Asimov postulated that an advanced robot might consider the prevention of harm to humanity a higher priority than the prevention of harm to an individual. The Zeroth reads: "A robot may not injure humanity or, through inaction, allow humanity to come to harm, and thus the Modified First Law becomes: "A robot may not injure a human being or, through inaction, allow a human being to come to harm, except where that would conflict with the Zeroth Law." Asimov himself questioned what we as humans do to all other living things and the planet we live on. VIKI echoed his notion by trying to protect mankind from mankind.

The movie even goes further. It ponders some other Asimovian philosophical topics: nanotechnology — the symbiosis of man and machine, robopsychology (have I mentioned that already?), robotic law (e.g., Who will be responsible for a robot's behavior — if a dying human asks his robotic aide to end his pain and suffering, will this form of euthanasia be acceptable? What laws will dictate robots hurting humans, destroying property, etc.), AI and machine consciousness, emotions, creativity, free will, and unanticipated behaviors in intelligent machines. These themes take *I, Robot* beyond the realm of science fiction.

Just recently, for example, Sony announced it's going to try to link Qrio (How is this pronounced? Might the public think it's "Queerio?" Now there's the name of a robot model type that makes them want to go out and buy one immediately!) to a grid of 250 computers so that it can make decisions on its own vs. responding to given instructions.

These social implications of robotics are brought to the masses by the movie *I, Robot*. The public (By the public, I am not referring to the erudite group of robo-enthusiasts — I'm referring to those that buy the tabloids at the supermarket. Like it or not, the tabloid articles typify the interests of middle America.) is now reading the book, *I, Robot,* with Will Smith on the cover and robots are table talk. iRobot — the Massachusetts-based company that manufactures consumer, military, and research robots — has received a deluge of calls asking for the NS5 robot. The public has learned that there are no NS5s yet (in case you, too, were wondering), but they did find out about real robots — the Roomba and the Packbot.

Whether I agree with others' impressions of the movie or not, I see it as an opportunity to create awareness and to bring further interest into the field. When I heard others discussing the movie and the doom and gloom of a potential robot revolution, I reminded them that that's what people thought in 1968 after seeing Hal in *2001, A Space Odyssey*.

Thirty-six years later, computers haven't murdered us. Actually, it's the humans (hackers) who can do the most damage (and this just may be what robotic psychiatry is truly about — the people who design and use robots — but that's a whole other story). I was hoping that after *I, Robot*, buffs would go into their garages that night and think how they could build their own Nestor, much like robot fighting created a fervor when it was brought to the masses via national television.

Asimov, I feel, would have been quite proud of the film version of *I, Robot*. Asimov, not modest, would have loved that the title of his book — nearly 65 years after its birth — is emblazoned on the big screen and that it is on TV, radio, newspapers, magazine covers, and media everywhere.

Asimov was the first to view robots as useful tools, friends, and servants. He invented the word robotics and, in doing so, he influenced the evolution of a new field and their creators. He believed that the Three Laws — his most important contribution to science fiction (at the age of 20, by the way) — are necessary for human safety with regard to robots.

To this day, though safety has continually been an intrinsic factor in the design of robots, nothing has thus far replaced nor disproved the significance of the Three Laws. I think Asimov would have been thrilled with the movie's interpretation of the Three Laws and its ability to bring to the public so many present-day issues surrounding robotic technology. *I, Robot* is the materialization of Asimov's dream. **SV**

Advertiser Index

All Electronics Corp. 47
Budget Robotics 61
CrustCrawler 37
Cyberbotics 11
Eagle Tree Systems 24
Hack-a-Sapien Contest 71
Hobby Engineering 57
Jameco 83
Lynxmotion, Inc. 15, 65
Net Media 2
Parallax, Inc. ....Back Cover
PCB123/PCBexpress 3
PCB Fab Express 7
Pololu Robotics & Electronics 27
Robodyssey Systems LLC 55
Robotic Trends 40
Rogue Robotics 13
Solarbotics 51
Solutions Cubed 12
Sozbots 33
Supercircuits 25
Surplus Sales of Nebraska 47
Team Delta 79
Technological Arts 32
Tetsujin 2004 8
Vantec 46
Yost Engineering, Inc. 21
Zagros Robotics 47

Lowest Prices Guaranteed!

Greatly Expanded Line of Quality Electronic Components

Celebrating 30 Years

Over 1,400 New Products Added

Semiconductors
Over 1,000 of the industry's most popular, ranging from Analog to Zilog.

Power Products
Your source for high quality power, whether you need medical grade or commercial grade.

NTE Replacement Semiconductors
The top 550 NTE quality products. All in-stock at unbeatable prices

Intel CPU's & Motherboards
We've assembled the top proven performers in microprocessors & motherboards to cover your requirements.

It's Worth Your Time To Shop Jameco Again!

- We Guarantee the Lowest Prices
- Up to 64% Savings from List Price
- Over 99% of Products In Stock
- Product Offering Expanded 50%

These are just some of the latest additions!
Log on today for so much more
www.Jameco.com/SR1

Get your FREE catalog today!
Just mention VIP # SR1

1-800-831-4242

JAMECO® ELECTRONICS
Brand Names & Generic Equivalents
Quality Components For Less

The Best Values
We search the world to find you the best values!

The Lowest Prices
We guarantee the lowest price in any catalog.

Greatly Expanded Products
We've added more than 50% new products in less than a year.

Immediate Availability
Over 99% of products are in stock for same day shipping.

Circle #33 on the Reader Service Card.

A Low-Cost Accelerometer

The **MEMSIC 2125 (#28017; $29.00)** is a low cost, dual-axis thermal accelerometer capable of measuring dynamic acceleration (vibration) and static acceleration (gravity) with a range of ±2 g. MEMSIC's unique thermal sensing technology fundamentally reduces accelerometer cost while enhancing system performance and reliability. The MEMSIC 2125 is electrically compatible with other popular accelerometers which will make integration into your existing applications simple.

Some of the MEMSIC 2125's features:

- Measure 0 to ±2 g on either axis; less than 1 mg resolution
- Fully temperature compensated over 0° to 70° C range
- Simple, pulse output of g-force for X and Y axis
- Analog output of temperature (TOut pin)
- Low current operation: less than 4 mA at 5 vdc

Possible BASIC Stamp® module/MEMSIC 2125 applications include:

- Dual-axis tilt sensing for autonomous robotics applications
- Single-axis rotational position sensing
- Movement/Lack-of-movement sensing for alarm systems
- R/C hobby projects such as autopilots

MEMSIC

MEMSIC (www.memsic.com) provides the 2125 in a surface-mount format. Parallax mounts the circuit on a through-hole providing all I/O connections so it can easily be inserted on a breadboard or through-hole prototype area.

MEMSIC is a registered trademark of MEMSIC, Inc. BASIC Stamp is a registered trademark of Parallax, Inc.

Order online at **parallax.com** or call our Sales Department toll-free at **888-512-1024** (Mon-Fri, 7 am to 5 pm, PST).

www.**parallax**.com

PARALLAX

Circle #106 on the Reader Service Card.